UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA
UNAN - MANAGUA

FACULTAD REGIONAL MULTIDISCIPLINARIA MATAGALPA

FAREM - MATAGALPA

TEMA:

ALGORITMOS DE ECUACIONES NO LINEALES POR POLINOMIOS DE ADOMIAN

TESIS PARA OPTAR AL GRADO DE DOCTOR EN MATEMÁTICA APLICADA

AUTOR:

MSC. DICSON ANTONIO MÉNDEZ LÓPEZ

TUTOR:

DR. IVÁN AUGUSTO CISNEROS DÍAZ

MATAGALPA, MARZO, 2023

¡A LA LIBERTAD POR LA UNIVERSIDAD!

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA

UNAN - MANAGUA

FACULTAD REGIONAL MULTIDISCIPLINARIA MATAGALPA

FAREM - MATAGALPA

TEMA:

ALGORITMOS DE ECUACIONES NO LINEALES POR POLINOMIOS DE ADOMIAN

TESIS PARA OPTAR AL GRADO DE DOCTOR EN MATEMÁTICA APLICADA

AUTOR:

MSC. DICSON ANTONIO MÉNDEZ LÓPEZ

TUTOR:

DR. IVÁN AUGUSTO CISNEROS DÍAZ

MATAGALPA, MARZO, 2023

¡A LA LIBERTAD POR LA UNIVERSIDAD!

1. AGRADECIMIENTOS

A Dios, por el conocimiento, la salud y las oportunidades.

A mi papá y mi mamá por su guía permanente y sus consejos.

Un agradecimiento especial a mi tutor, el Dr. Iván Augusto Cisneros Díaz, por la paciencia y la motivación en cada momento de la tesis.

A cada uno de los docentes que nos impartieron clases en el programa y que nos ayudaron en el crecimiento personal.

A mi hermano Huberth, mi crítico permanente.

A Ximena, por estar presente y apoyarme en todo el proceso de esta tesis.

2. CARTA AVAL

Dr. Iván Augusto Cisneros Díaz, docente titular del Departamento de Enseñanzas de las Ciencias de la Facultad de Educación e Idiomas, de la Universidad Nacional Autónoma de Nicaragua, UNAN-Managua.

CERTIFICA que la presente Monografía

"Algoritmos de Ecuaciones no Lineales por Polinomios de Adomian"

ha sido realizada bajo su dirección en el Departamento de Enseñanzas de las Ciencias por el Máster

Dicson Antonio Méndez López

y constituye su tesis para optar al grado de Doctor en Matemática Aplicada.

Y para que así conste, en cumplimiento con la normativa vigente, autoriza su presentación ante la Coordinación del Programa de Doctorado y Maestría, para que pueda ser tramitada su lectura y defensa pública.

Tutor de Tesis

Dr. Iván Augusto Cisneros Díaz

Nicaragua, Matagalpa, Marzo, 2023.

3. RESUMEN

Este trabajo se realizó con el objeto de mejorar y optimizar los procesos iterativos de aproximación de soluciones a ecuaciones no lineales. El método de Newton es un algoritmo iterativo que permite resolver estos tipos de ecuaciones. La investigación desarrollada consistió en encontrar nuevos esquemas y métodos iterativos equivalente o superiores en el número de iteraciones al método de Newton. Esta tesis plantea la relación natural que existe entre los Métodos de Descomposición de Adomian y la Técnicas Iterativa Variacional, estableciendo los vínculos matemáticos desarrollados en ambas esferas del conocimiento. Para las demostraciones de los nuevos esquemas y métodos iterativos se basó en el esquema de los polinomios de Adomian y luego se combinó con las técnicas iterativas variacional, obteniéndose de esta manera, nuevas fórmulas iterativas de cálculo de raíces de ecuaciones no lineales. En todos los casos se utilizó una función auxiliar familia de las funciones exponenciales, ya que tienen la particularidad de ser funciones C^{∞}. El objetivo principal es demostrar dichas fórmulas iterativas y mostrar que la teoría matemática desarrollada en este campo científico, están fundamentadas teóricamente y analíticamente por métodos y procedimientos lógicos, que permiten desarrollar nuevos esquemas, métodos y técnicas iterativas. Los algoritmos son generados mediante los procedimientos de los polinomios de Adomian y la Técnica Iterativa Variacional. Este trabajo presenta treinta algoritmos nuevos que permiten encontrar las soluciones a ecuaciones no lineales en una cantidad menor de iteraciones que el método de Newton y por lo tanto son más eficientes que dicho método. Todos estos algoritmos fueron programados en el lenguaje de programación Python y se utilizó el paradigma de programación orientado a objetos (POO). Todos estos nuevos algoritmos presentan convergencia en dicha solución. Las ideas de este trabajo pueden extenderse para generar nuevos algoritmos con los Método de Abbasbandy y Cisneros en la búsqueda de algoritmos más eficientes.

Palabras Claves: Método de Newton, Polinomios de Adomian, Iteración Variacional, Algoritmos Iterativos.

4. ABSTRACT

This work is carried out in order to improve and optimize the iterative processes of approximation of solutions of non-linear equations. Newton's method is an iterative algorithm that allows solving these types of equations. The research carried out consisted of finding new schemes and iterative methods equivalent to Newton's method. This thesis raises the natural relationship that exists between Adomian Decomposition Methods and Variational Iterative Techniques, establishing the mathematical links developed in both spheres of knowledge. For the demonstrations of the new schemes and iterative methods, it was based on the Adomian Polynomial scheme and then combined with variational iterative techniques, obtaining in these ways new iterative formulas for calculating roots nonlinear equations. In all cases, a familiar auxiliary function of exponential functions was used, since they have the particularity of being C^{∞} functions. The main objective is to demonstrate these iterative formulas and to show that the mathematical theory developed in this scientific field are theoretically and analytically based by logical methods and procedures, which allow the development of new iterative schemes and techniques. The algorithms are generated by the procedures of the Adomian Polynomials and the Variational Iterative Technique. This work presents fifteen new algorithms that allow finding solutions to nonlinear equations in a smaller number of iterations than Newton's method and therefore are more efficient than said method. All these algorithms were programmed in the Python programming language and the object-oriented programming paradigm (OOP) was used. All these new algorithms present convergence in said solution. The ideas of this work can be extended to generate new algorithms with the Abbasbandy and Cisneros Methods in the search for more efficient algorithms.

Keywords: Newton's method, Adomian polynomials, Variational Iteration, Iterative Algorithms.

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA
UNAN - MANAGUA

FACULTAD REGIONAL MULTIDISCIPLINARIA MATAGALPA

FAREM - MATAGALPA

TEMA:

ALGORITMOS DE ECUACIONES NO LINEALES POR POLINOMIOS DE ADOMIAN

TESIS PARA OPTAR AL GRADO DE DOCTOR EN MATEMÁTICA APLICADA

AUTOR:

MSC. DICSON ANTONIO MÉNDEZ LÓPEZ

TUTOR:

DR. IVÁN AUGUSTO CISNEROS DÍAZ

MATAGALPA, MARZO, 2023

¡A LA LIBERTAD POR LA UNIVERSIDAD!

Índice

Índice de tablas

5. CAPÍTULO I: GENERALIDADES

5.1. Introducción

Es indudable que la Matemática genera un gran impacto en la sociedad actual, así lo atestigua el progreso descrito por los modelos matemáticos basados en teoría de funciones, ecuaciones lineales y no lineales, tópicos de álgebra lineal y álgebra abstracta. Así que el estudio de la teoría de ecuaciones es muy importante y como es comprensible, la generación de métodos numéricos para resolverlas constituye un reto importante para los matemáticos actuales.

El estudio de las ecuaciones es tan antiguo como la Matemáticas misma, pero a medida que la ciencia fue avanzando también se necesitaron ecuaciones más complejas, que no presentan una solución que se puede determinar mediante una simplificación algebraica. El método de Newton permitió generar soluciones a diversas ecuaciones, y existía al mismo tiempo otros métodos que presentaban convergencia en una cantidad de iteraciones similar al método de Newton.

El estudio de los polinomios de Adomian permitió a Abbasbandy desarrollar un algoritmo efectivo que mejora al método de Newton. Cisneros (2017) hizo importantes avances en este campo y logró desarrollar métodos numéricos sumamente precisos que requerían menos iteraciones que el método de Newton.

En muchos casos estas ecuaciones no tienen una solución exacta y se necesitan métodos aproximados como el método de Newton que permite obtener aproximaciones a las raíces de una ecuación de una forma práctica y eficiente, pero que puede mejorarse de distintas formas, como se ha demostrado en otros trabajos.

Es necesario destacar que existen diversos procesos matemáticos que permiten resolver ecuaciones no lineales, sin embargo, existe la posibilidad de estudiar el análisis de convergencia de dichos métodos, mediante técnicas iterativas, las cuales permiten optimizar de manera eficiente la solución numérica de dichas ecuaciones.

Esto justifica una investigación que permita diseñar nuevos algoritmos basado en la técnica de descomposición de los polinomios de Adomian y de las Técnicas Iterativas Variacionales, que permita resolver ecuaciones no lineales en la línea de investigación del Álgebra y Análisis Matemático, y de esta forma establecer un conjunto de algoritmos que permitan generalizar el método de Newton y codificar estos algoritmos en programas computacionales, tal como el lenguaje de programación Python.

Esta tesis se divide en varios capítulos, entre ellos:

El capítulo I, se refiere a generalidades de la tesis desarrollada, entre ellos se aborda lo referente a: introducción, planteamiento del problema, antecedentes, justificación y objetivos a alcanzar en la investigación. En ellos se muestra la importancia y relevancia de la investigación, implicaciones prácticas, valor teórico, utilidad metodológica y viabilidad de la investigación.

El capítulo II, muestra el estado del arte referido al proceso de investigación desarrollado, con el objetivo de presentar la amplitud y el tratamiento de los polinomios de Adomian, sus formas genéricas de cálculos de dichos coeficientes y sus variantes con el desarrollo de funciones por medio de series de Taylor.

El capítulo III, presenta un abordaje de los conceptos generales sobre la teoría de los métodos iterativos, enfatizando en aquellas definiciones y teoremas generales necesarios para la investigación, también presenta algunos algoritmos para encontrar las raíces de ecuaciones no lineales, con el propósito de mostrar la amplitud de las diversas técnicas numéricas y poder dirigir el estudio a la continuación de los conocimientos que se han alcanzado. También se presenta un listado de definiciones, propiedades y teoremas relativos a los métodos iterativos, destacando el método de Newton.

El capítulo IV, muestra el desarrollo de los polinomios de Adomian, resaltando nuevos esquemas iterativos y obteniendo nuevas fórmulas iterativas para el hallazgo de las raíces reales de ecuaciones no lineales. Se analiza la técnica de Descomposición de Adomian y sus diversas variantes.

En el capítulo V, se muestra el análisis de resultados de los 30 algoritmos que superan o son equivalentes al método de Newton, de acuerdo al número de iteraciones necesarias para alcanzar la raíz real deseada. Es necesario señalar que la selección de estas funciones bases, obedecen a criterios de comparación que se efectuaron entre las aproximaciones numéricas presentadas en los artículos científicos consultados y los obtenidos por los diversos programas que se codificaron para el análisis de la información.

Por último, se presentan las conclusiones generales de la investigación. La programación de todos los algoritmos fue codificada en el lenguaje de alto nivel Python, para este fin, se utilizó técnicas de POO (Programación Orientada a Objetos), es decir, programación que emplea definición de clase.

5.2. Planteamiento del Problema

Las ecuaciones no lineales son usadas en diversas ciencias como modelos para resolver problemas, entre estas ciencias están la Física, la Astronomía, Economía, Química, y algunas más, por esta razón se deben proponer nuevos métodos numéricos que permitan encontrar soluciones en un menor número de iteraciones.

La tecnología ha destruido los viejos paradigmas relacionados con la Matemática, el trabajo manual ha sido sustituido por programas computacionales, donde la creación de un método o un conjunto de métodos numéricos permitan obtener una solución en una iteración menos que otros algoritmos existentes o que tenga un menor coste computacional significa una ganancia significativa en eficiencia.

Por lo antes expuesto surge el interés en desarrollar métodos numéricos basados en el método de Newton que permitan resolver ecuaciones no lineales, encontrando variantes de este método mediante la técnica de descomposición de polinomios de Adomian e iteración variacional.

5.3. Antecedentes

Las ecuaciones no lineales son importantes en las ciencias, por esta razón, se deben proponer nuevos métodos numéricos que permitan encontrar soluciones que se realicen de forma sencilla y que puedan encontrar aproximaciones más exactas que los métodos existentes actualmente.

La computación ha destruido los viejos paradigmas relacionados con la Matemática, el trabajo manual ya ha sido sustituido por programas computacionales, donde la creación de un método o un conjunto de métodos numéricos que se obtenga en una iteración menos que otros existentes o que tenga un menor coste computacional significa una ganancia significativa en eficiencia.

Por lo antes expuesto surge el interés en desarrollar métodos numéricos óptimos y, eficientes que permitan resolver ecuaciones no lineales mediante la técnica de descomposición de polinomios de Adomian y otras técnicas iterativas multipasos.

Los métodos para resolver ecuaciones son antiguos. Muchos de estos métodos son exactos, por ejemplo, para las ecuaciones de la forma lineal y polinomial $ax + b = 0$ presenta soluciones exactas mediante la ecuación $x = -\frac{b}{a}$. Pero, conforme las ecuaciones se hacen más complejas ya sea en orden o no son lineales (combinaciones de funciones transcendentes), los procesos empiezan a aumentar su complejidad y no existen fórmulas para resolver de forma analítica dichas ecuaciones. En este caso, surgen los métodos iterativos que proporcionan métodos aproximativos numéricos..

A partir de la revisión de diversos artículos científicos donde se aborda el método de Newton y mediante diversas técnicas numéricas, se puede concluir que es un tema donde se puede profundizar estableciendo algoritmos diferentes a los existentes y que tengan una convergencia equivalente o mayor.

Los diversos estudios analíticos, los métodos de un paso y multipasos pueden establecer una convergencia de hasta un cuarto orden a costa de determinar derivadas de orden superior.

En los análisis de los esquemas de Halley, Chebyshev, Convexidad Logarítmica, Potra-Ptak, Jarratt, Ostrowski y los métodos de King se puede apreciar la evolución de los diferentes métodos numéricos relacionados con el esquema propuesto por Newton.

Nivel Internacional

Los algoritmos propuestos en el artículo A Variant of Newton's Method with Accelerated Third-Order Convergence presentado por Weerakoon y Fernando (1998, p.88), marcan el inicio del diseño de métodos iterativos multipasos, desarrollado desde el teorema de integración del punto medio, desarrollando algoritmos bipasos. Es destacable lo fructífera que ha sido esta forma de mejorar el método de Newton.

Abbasbandy (2003, p.887), en un artículo llamado Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method, compara sus dos algoritmos presentados con otros métodos numéricos y con diferentes ecuaciones mostrando la viabilidad del uso de la descomposición de Adomian para generar nuevos esquemas que superen al método de Newton.

Chun, C. (2005, p.1559), publicó Iterative Methods Improving Newton's Method by the Decomposition Method, obra que profundiza en métodos numéricos basados en la Descomposición de Adomiam combinada con los métodos multipasos.

Shah (2012), presenta su tesis doctoral titulada Variational Iteration Technique and Numerical Methods for Solving Nonlinear Equations, donde genera algoritmos para aproximar raíces de ecuaciones no lineales mediante la técnica de variación iteracional.

Muchos algoritmos se han producido en los últimos veinte años, bajo distintas líneas de investigación, cada uno con sus propiedades y estructuras diferentes con el fin de optimizar el método de Newton.

A nivel nacional

A nivel nacional Cisneros (2017), publica su tesis Algoritmos basados en los polinomios de Adomian e Interación Variacional para la resolución de ecuaciones no lineales, generando algoritmos óptimos que disminuían el número de iteraciones del método de Newton.

Herrera y Cisneros (2022), presentan tres algoritmos basados en iteración variacional en la revista de la Universidad Nacional Autónoma de Nicaragua, en la facultad de Estelí, Nicaragua, en un artículo llamado, Algoritmización para la resolución de ecuaciones no lineales mediante la técnica de iteración variacional.

5.4. Justificación

Este trabajo pretende desarrollar diferentes métodos numéricos basados en la descomposición de los polinomios de Adomian para resolver ecuaciones no lineales y que faciliten la obtención de sus raíces reales, logrando que diferentes profesionales que usan ecuaciones no lineales aplicadas a sus diferentes trabajos y campos de estudio resuelvan sus problemas óptimamente. Se debe recordar que muchas ciencias expresan sus teorías, explicaciones y predicciones en forma de ecuaciones no lineales, por ejemplo, la Física, la Astronomía, la Economía, entre otros.

La creación de métodos numéricos incide en la generación de esquemas iterativos y por ende de nuevos algoritmos computacionales. Diferentes artículos de investigación destacan la importancia teórica y práctica de la descomposición de los polinomios de Adomian.

Los polinomios de Adomian permitieron ampliar y diversificar el método iterativo de Newton y a partir de este hecho, se ha abierto una línea de investigación que ofrece nuevos horizontes del proceso investigativo matemático.

Es posible realizar esta investigación porque se cuenta con las herramientas tecnológicas necesarias y también con la existencia de diferentes paradigmas de programación que permiten desarrollar el estado de la presente investigación.

5.5. Objetivos

5.5.1. Objetivo General

Diseñar nuevos esquemas iterativos basados en el Método de Newton mediante la técnica de descomposición de los polinomios de Adomian para resolver ecuaciones no lineales.

5.5.2. Objetivos Específicos

1. Generalizar el método de Newton, basado en diferentes enfoques matemáticos y métodos aproximados para la solución de ecuaciones no lineales.

2. Diseñar nuevos algoritmos iterativos mediante la técnica de descomposición de los polinomios de Adomian y la Técnica Iterativa Variacional para la solución de ecuaciones no lineales.

3. Programar los nuevos algoritmos mediante el lenguaje de programación Python, utilizando el paradigma de programación orientada a objetos (POO).

6. CAPÍTULO II: ESTADO DEL ARTE

Las ecuaciones se han usado para resolver diferentes problemas específicos de las ciencias aplicadas, sin embargo, existen algunas ecuaciones que no cuentan con un proceso algebraico para resolverse, a estas se les llama no lineales, por ejemplo, se usan en áreas como la astronomía, la inteligencia artificial y la robótica.

Una forma simple de ecuaciones no lineales son las de una variable de la forma $f(x) = 0$, con una raíz real α tal que $f(\alpha) = 0$. Para resolver esta ecuación se han propuesto métodos iterativos, estos se clasifican en de un solo paso o multipasos.

Un método popular para resolver una ecuación como:

$$f(x) = 0$$

es el método iterativo de Newton, que se expresa de la siguiente forma

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

Donde k es un número entero mayor o igual que cero. Si $k \to \infty$ entonces $x_{k+1} \to \alpha$ donde α es la solución de $f(x) = 0$.

El método de Newton requiere solo encontrar la primer derivada, a diferencia de otros métodos evita el uso de derivadas de orden superior. Se debe aclarar que la función debe ser monótona y la raíz debe ser simple es decir que $f'(\alpha) \neq 0$. El orden de convergencia del método de Newton es dos, presenta una forma muy simple y generalmente requiere pocas iteraciones, en especial si se toma un valor para empezar el esquema cercano a la solución α. A veces este método es conocido como Newton-Raphson como en el libro de Cálculo de Zill (1996, p.171) y otros documentos formales y reconocidos, para sintetizar en este trabajo únicamente será mencionado como Método de Newton.

Los trabajos de distintos matemáticos para mejorar el orden de convergencia condujeron a otros métodos, como los esquemas de Halley, Chebyshev, House Holder, entre otros, que presentan una convergencia cúbica y el operador de convexidad logarítmica el cual viene dado por la siguiente expresión:

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

En esta línea se presenta el método de Euler-Chevishev con la siguiente forma citado por Díaz-Barrero y Grau (2003, p.3)

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \left(1 + \frac{f(x_k) f''(x_k)}{[f'(x_k)]^2}\right), \ k = 0, 1, 2, ...$$

Es decir

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}\left(1 + L_f(x_k)\right), \ k = 0, 1, 2, ...$$

Según Amat, Busquiera y Gutiérrez (2003, p.199) el método de Halley viene dado por

$$x_{k+1} = x_k - \frac{2}{2 - L_f(x_k)}\frac{f(x_k)}{f'(x_k)}, \ k = 0, 1, 2, ...$$

O también por

$$x_{k+1} = x_k - \frac{2}{2 - \dfrac{f(x_k)\, f''(x_k)}{[f'(x_k)]^2}}\frac{f(x_k)}{f'(x_k)}, \ k = 0, 1, 2, ...$$

Amat et al (2003, p.199) señalan que el método de Chevishev es

$$x_{k+1} = x_k - \left(1 + \frac{1}{2}L_f(x_k)\right)\frac{f(x_k)}{f'(x_k)}, \ k = 0, 1, 2, ...$$

Lo que sería en la forma extendida

$$x_{k+1} = x_k - \left(1 + \frac{1}{2}\frac{f(x_k)\, f''(x_k)}{[f'(x_k)]^2}\right)\frac{f(x_k)}{f'(x_k)}, \ k = 0, 1, 2, ...$$

Otro esquema es llamado super Halley y según Amat et al (2003, p.200) es

$$x_{k+1} = x_k - \left(1 + \frac{1}{2}\frac{1}{1 - L_f(x_k)}\right)\frac{f(x_k)}{f'(x_k)}, \ k = 0, 1, 2, ...$$

Usando otros razonamientos Stepenson, citado por Kumar, Singh y Srivastava (2013, p.335), brinda el siguiente algoritmo donde la primer derivada del método de Newton es sustituida por una aproximación:

$$x_{k+1} = x_x + \frac{f^2(x_k)}{f(x_k + f(x_k)) - f(x_k)}, \ k = 0, 1, 2, ...$$

Algunas veces este método se le añade una constante en su estructura

$$x_{k+1} = x_x + \frac{\beta f^2(x_k)}{f(x_k + \beta f(x_k)) - f(x_k)}, \ k = 0, 1, 2, ...$$

En esta forma es trabajado por Bahgat y Hafiz (2012, p.38).

Las limitaciones de los métodos punto a punto llevan, 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. Un esquema de orden de convergencia tres, propuesto por Traub, citado por Cisneros (2017, p.10) el cual su expresión iterativa 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, es decir

$$\begin{aligned} y_k &= x_k - \frac{f\left(x_k\right)}{f'\left(x_k\right)} \\ x_{k+1} &= y_k - \frac{f\left(y_k\right)}{f'\left(y_k\right)} \quad , \quad k = 0, 1, 2, ... \end{aligned}$$

Los métodos clásicos generaron la familia de algorítmos de Chebyshev-Halley, desarrollándose después del esquema de Traub. Jarratt, Ostrowski y los métodos de King, los cuales no hacían uso de la evaluación de la derivada segunda, todo ellos con orden de convergencia cuatro.

El método de Potra-Ptak puede escribirse de la siguiente forma según Thukral (2016, p.16):

$$\begin{aligned} y_k &= x_k - \frac{f\left(x_k\right)}{f'\left(x_k\right)} \\ x_{k+1} &= y_k - \frac{f(x_n) + f\left(y_k\right)}{f'\left(y_k\right)} \quad , \quad k = 0, 1, 2, ... \end{aligned}$$

Diversos autores analizaron métodos iterativos en su convergencia local mediante la determinación de regiones donde se aseguraba su convergencia, principalmente en funciones monótonas. Como el trabajo de Weerakoon, Fernando y Ozban citado por Kou, Li y Wang (2006, p.2)

$$x_{k+1} = x_k - \frac{2f(x_k)}{f'(x^*_{k+1}) + f'(x_k)}$$

Donde

$$x^*_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad , \quad k = 0, 1, 2, ...$$

Este método fue presentado originalmente en un artículo llamado A Variant of Newton's Method with Accelerated Third-Order Convergence presentado por Weerakoon y Fernando (1998, p.88). Ozban (2004, p.677) creó a partir de estos métodos un conjunto de variantes de este algoritmo de dos pasos mediante la regla de integración del punto medio. Siendo esta línea de investigación una de las más estudiadas por los matemáticos, y donde más resultados se han producido.

Una variante de estos métodos son la familia de soluciones dada por el siguiente esquema iterativo presentado por Subrahmanyam y Vijesh (2007, p.361)

$$x_{k+1} = x_k + (\lambda f'_{x_n}\left(x_k\right) + (1+\lambda) f'_{z_n}\left(x_k\right))^{-1} f\left(x_k\right), \ k = 0, 1, 2, ...$$

El cual se puede simplificar como

$$x_{k+1} = x_k + \frac{f\left(x_k\right)}{\lambda f'\left(x_k\right) + (1+\lambda) f'\left(z_k\right)}, \ k = 0, 1, 2, ...$$

Donde λ es un número entre cero y uno.

Un modelo multipaso es el de Ostrowski (publicado en la década de los setentas), citado por Koua, Lib y Wang (2006, p.154) en un trabajo sobre la convergencia de este método, en síntesis, este algoritmo presenta la siguiente forma:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - \frac{f(y_k)(x_k - y_k)}{f(x_k) - 2f(y_k)} \quad , \quad k = 0, 1, 2, ... \end{aligned}$$

Este es un caso particular de un modelo general como lo señala Bi, Ren y Wu (2009, p.105) esquema basado en la familia de métodos de King nombrado de esta forma por los autores Petkovic y Thukral (2010, p.2278).

Chun (2006, p.4) presentó diversos algoritmos multipasos como:

$$\begin{aligned} y_{k+1} &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_k &= x_k - 2\frac{f^2(x_k) + f^2(y_k)}{f'(x_k)(f(x_k) + f(y_k))} \quad , \quad k = 0, 1, 2, ... \end{aligned}$$

Un método de tres pasos es el que desarrollan Cordero, Torregrosa y Vassileva (2011, p.3190)

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ z_k &= x_k - \frac{f(x_k)}{f'(x_k)} \frac{f(x_k) - f(y_k)}{f(x_k) - 2f(y_k)} \\ x_{k+1} &= x_k - \frac{f(z_k)}{f'(x_k)} \left[\left(\frac{f(x_k) - f(y_k)}{f(x_k) - 2f(y_k)} \right)^2 + \frac{f(z_k)}{f(y_k) - \alpha f(z_k)} + G(\mu_k) \right] \end{aligned}$$

Donde α es un número real y $G(\mu_k)$es una función de valores reales y $\mu_k = \dfrac{f(z_k)}{f'(x_k)}$y $k = 0, 1, 2, ...$

Noor presentó algoritmos en varios artículos en colaboración con autores como ejemplo, en el artículo publicado por Noor, Noor y Aftad (2012, p.870) se encuentra:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - 2\frac{f(x_k)}{f'(x_k)} - (y_n - x_n)\frac{f(y_k)}{f'(x_k)} \quad , \quad k = 0, 1, 2, ... \end{aligned}$$

Como se puede observar la cantidad de método multipasos, así como de familias de métodos numéricos producidos por diversos autores a lo largo de los años. Un importante trabajo consiste en deducir nuevos métodos numéricos a partir de los métodos comunes y conocidos en especial la literatura es amplia en el caso del método de Newton, pero existen trabajos relacionados con otros métodos como los mencionados anteriormente o aún más reciente.

La necesidad de crear métodos numéricos de orden mayor ha generado técnicas diversas, como lo cita Cisneros (2017) "la composición de métodos conocidos (básicamente Newton) y la posterior eliminación de algunas de las evaluaciones funcionales añadida en el proceso" (p.10). Las técnicas más efectivas para este proceso se basan en la generalización de métodos hacia el estudio de nuevas familias, que pueden ser mediante la variación iteracional o el método de polinomios de Adomian, también existen métodos de tres o cuatro pasos que permiten acelerar la convergencia de los métodos, o los métodos basados en la estructura del esquema de Potra-Ptak.

El método de Iteración variacional es un método propuesto para la resolución de ecuaciones diferenciales que consiste en que "una función auxiliar es construida por un multiplicador general de Lagrange que puede generar una vía óptima a un proceso variacional" (Ganji, Nourollahi y Rostamian, 2006, p.178), el método permite la resolución de ecuaciones diferenciales aplicadas a la ingeniería de una forma más sencilla, por tal razón, puede ser usado con éxito para encontrar algoritmos óptimos para encontrar raíces de ecuaciones.

Shah (2012) en su tesis doctoral usó la técnica de iteración variacional para encontrar algoritmos que disminuyen las iteraciones. El usó ecuaciones auxiliares de igual forma que Noor y el desarrollo y sustitución de productos de la forma: $H(x) = x + \lambda[f(x)g(x)]$. Las ecuaciones auxiliares son de la familia exponencial como $g(x) = e^{-\alpha x}$, $g(x) = e^{-\frac{\alpha}{f'(x)}}$, $g(x) = e^{-\alpha\frac{f(x)}{f'(x)}}$.

El método de descomposición de Adomian consiste en encontrar una serie infinita de funciones, que es aproximadamente igual a una función dada. Este proceso también fue usado para aproximar y mejorar el método de Newton, el método de Abbasbandy se basa en este proceso, Chun (2003) también modificó el método de Newton a partir de la descomposición de Adomian.

A nivel nacional los métodos numéricos fueron estudiados por Cisneros (2017), en su tesis doctoral presentó diversos algoritmos basados en la técnica de iteración variacional, entre los que se pueden encontrar el siguiente algoritmo con $k = 0, 1, 2, ... y$ α es un número real.

$$
\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\
x_{k+1} &= y_k - \frac{f(y_k)}{f'(x_k) - \alpha f'(x_k)}
\end{aligned}
$$

Además Cisneros (2017) trabajó algoritmos basados en los polinomios de Adomian, y otros con la técnica de iteración variacional.

7. CAPÍTULO III: MARCO TEÓRICO

7.1. Conceptos Generales

La definición de función es muy importante, así que según Swokowski y Cole (2009, p.178)

Definición 1 *Una función f de un conjunto D a un conjunto E es una correspondencia que asigna a cada elemento x de D exactamente un elemento y de E.*

Existen diferentes clasificaciones de funciones, Montalvo (2003, p.69) señala que:

Definición 2 *Sea $f : U \subset E \to F$ donde U un conjunto abierto de E. Diremos que f es de clase C^1sobre $A \subset U$, lo que denotaremos por $f \in C^1(A)$ si (i) f es diferenciable en cada punto x de A. (ii) La aplicación $Df : x \to Df(x)$ es continua en A.*

Puede decirse que una función es de clase C^1 si es derivable en un intervalo, (para este caso el intervalo es un subconjunto de los números reales) y su derivada es continua en dicho intervalo. Se puede observar que no se requiere la continuidad en todo el dominio D de la función f.

En general, si una función es de C^1, indica que posee una primer derivada, una función de C^2, que posee una segunda derivada, C^3, indica que posee una tercera derivada, y así sucesivamente, Azagra (2018, p.1), manifiesta

Definición 3 *Sean U un abierto de $\mathbb{R}^n$, y $k \in N$. Se dice que una función $f : U \subseteq \mathbb{R},\ n \to \mathbb{R}$ es de clase C^k, y se escribe $f \in C^k(U)$, si todas sus derivadas parciales de orden k existen y son continuas en U. Diremos que $g : U \to \mathbb{R}^m$ es de clase C^k, y escribiremos $g \in C^k(U, \mathbb{R}^m)$, si cada función componente de g es de clase C^k.*

Estas derivadas pueden ir desde la primer derivada, hasta una derivada cualquiera, se puede expresar que "f es de C^∞, y escribimos $f \in C^\infty(U)$ si f es de clase C^k para cada $k \in \mathbb{N}$" (Azagra, 2018, p.1).

Las sucesiones en los números reales Bartle y Sherbert (2004, p.85) afirman que:

Definición 4 *Una sucesión de números reales (o sucesión en $\mathbb{R}$) es una función en el conjunto $\mathbb{N}$ de los números naturales cuyo codominio está contenido en el conjunto de los números reales. A estos números obtenidos se les llaman elementos, valores o términos de la sucesión.*

Existen diversas clasificaciones de las funciones, a una sucesión con infinitos términos se le llama sucesión infinita, por consiguiente, una función con una cantidad específica de elementos se le considera sucesión finita.

El concepto de Sucesión infinita convergente es fundamental en un esquema Iterativo, Rudin (1980, p.50) señala que:

Definición 5 *Dada una sucesión $\{p_n\}$ en un espacio métrico (E,d) converge hacia el elemento $p \in X$, con las siguientes propiedades: para cada $\varepsilon > 0$ existe un número entero N tal que para todo $n > N$ se verifica que $d(p_n, p) < \varepsilon$. En forma simbólica*

$$\lim_{n\to\infty} p_n = p \Longleftrightarrow \forall\ \varepsilon > 0,\ \exists\ N \in \mathbb{N}\ \textit{tal que}\ \ n \geq N\ \textit{implica que}\ \ \ |p_n - p| < \varepsilon$$

La notación $\lim_{n\to\infty} p_n = p$ significa que la sucesión $\{p_n\}$ converge a p. A esta sucesión se le llama convergente, de lo contrario la sucesión es divergente. Y la letra d se le llama distancia.

Los resultados del proceso iterativo forman una sucesión convergente de Cauchy, la cual según Rudin (1980, p.55) puede definirse así:

Definición 6 *Dada una sucesión infinita de elementos $\{p_n\}$ del espacio métrico (E,d) se dice que la sucesión es una sucesión de Cauchy, si para cualquier valor $\varepsilon > 0$ hay un entero N tal que $d(p_n, p_m) < \varepsilon$ si $n > N$ y $m > N$.*

Para determinar la convergencia de las sucesiones de Cauchy no requieren encontrarles un límite.

Dado que las ecuaciones son ejemplos de punto fijos de aplicaciones es necesario establecer la siguiente definición, citando a Cisneros (2017, p.11):

Definición 7 *Sea g una aplicación definida en un espacio métrico (E,d) y con valores en el mismo espacio métrico, se denomina punto fijo de la aplicación g a todo elemento x^* de E tal que $x^* = g(x^*)$. Geométricamente un punto fijo representa el punto de intersección de la curva $y = g(x)$ con la recta $y = x$.*

La existencia de las aplicaciones lipschitziana permiten realizar demostraciones más sencillas de la convergencia de sucesiones, citando a Cisneros (2017, p.12):

Definición 8 *Sean (E,d) y (V,d^*) dos espacios métricos y sea $g : E \to V$ una aplicación definida en E y con valores en V. Se dice que g es una aplicación lipschitziana de razón k cuando existe una constante real $k > 0$ tal que:*

$$d^*\left(g(x), g(y)\right) \leq k\ d(x,y),\ \forall\ x, y \in E$$

A la menor constante k que verifica la condición anterior se la denomina constante de Lipschitz (o razón) de la aplicación. Hay un tipo de funciones lipschitzianas de interés especial, las funciones contractivas, que son funciones lipschitzianas para las cuales k toma valores $0 < k < 1$.

Las aplicaciones lipschitziana presentan las siguientes características, según Cisneros (2017, p.12):

1. Toda aplicación lipschitziana definida en (E, d) y con valores en (V, d^*) es una aplicación continua en todo E.

2. Si la constante de Lipschitz es inferior a uno se le denomina contracción sobre E.

3. Toda contracción definida sobre un espacio métrico completo admite un único punto fijo.

4. La distancia que separa x_n de la solución $x^* = (x_\infty)$ podría estimarse mediante

$$d\left(x_n, x^*\right) = \frac{k^n}{1-k} d\left(x_0, x_1\right)$$

5. La importancia del valor semilla x_0 para nuestro método iterativo.

6. La importancia de k sea próxima a cero, para disminuir la cantidad de iteraciones.

7. La distancia entre dos aproximaciones x_{n-1} y x_n es $\dfrac{k}{1-k} d\left(x_{n-1}, x_n\right)$

Los esquemas iterativos tal como el método de Newton están diseñados para resolver ecuaciones. Se entiende ecuación como una igualdad que contiene al menos una variable. La solución exacta de una ecuación según Cisneros (2017, p.15) en los números reales puede definirse de la siguiente manera:

Definición 9 *El número $\alpha \in \mathbb{R}$ se dice una solución de la ecuación si se verifica que $f(\alpha) = 0$, es decir, si α es una raíz de la ecuación f.*

El error es objeto de análisis en un método numérico, y se presentan diferentes definiciones y clasificaciones, Burden, Faires y Burden (2017, p.14) definen el error así:

Definición 10 *Sea x una aproximación a α. El error real es $x - \alpha$, el error absoluto es $|x - \alpha|$ y el error relativo es $\dfrac{|x-\alpha|}{|\alpha|}$ siempre y cuando $\alpha \neq 0$.*

De igual forma, el error en una solución obtenida según Cisneros (2017, p.15) es

Definición 11 *El error absoluto, representado por e_n, se obtiene como el valor absoluto de la diferencia entre el valor exacto x y la aproximación a este valor α. Esto es*

$$e_n = |x - \alpha|$$

Noor (2007, p.130) define el error de forma similar a la siguiente

Definición 12 *Sea una ecuación $f(x) = 0$, el error e en la iteración n de un esquema iterativo es*

$$f(x_{n)} = e$$

La necesidad del concepto de error radica en que se obtendrán aproximaciones a las soluciones α de las ecuaciones, así que se puede definir el error relativo de un método numérico, de la siguente manera

Definición 13 *El error relativo de un método numérico e_r dadas dos aproximaciones sucesivas x_n y x_{n-1} de forma que $x_n \to x_{n-1} \to \alpha$ es*

$$e_r = \left| \frac{x_n - x_{n-1}}{x_n} \right|$$

En general, este error será un valor cercano a cero, y su importancia radica en que será un criterio de parada en los algoritmos de aproximación de soluciones de ecuaciones no lineales.

Los métodos numéricos se clasifican mediante el orden de convergencia, Verma (2016, p.103) señala lo siguiente,

Definición 14 *Dada $f(x)$ la cual es una función real con una raíz simple α y $\{x_n\}$ es una sucesión de números reales que converge a α. Entonces el orden de convergencia de la sucesión es p si existe un número real positivo p, si existe una constante $\lambda > 0$ tal que*

$$\lim_{j\to\infty} \frac{|e_{j+1}|}{|e_j|^p} = \lim_{j\to\infty} \frac{|x_{j+1} - \alpha|}{|x_j - \alpha|^p} = \lambda$$

la constante λ se llama constante asintótica del error.

Los órdenes de convergencia probables son, $p = 1$ es una convergencia lineal, si $p = 2$ se le llama convergencia cuadrática, si $p = 3$ es una convergencia cúbica.

Según Cisneros (2017, p.16) otra definición de orden de convergencia es:

También se puede definir al orden de convergencia de la sucesión $\{x_n\}_{n=0}^{\infty}$ como el número $p \geq 1$ tal que

$$|e_{n+1}| \leq C\,|e_n|^p \quad \text{para algún } C > 0 \quad y \quad 0 < C < 1$$

Como es evidente un orden de convergencia mayor es mejor que un orden de convergencia menor.

El orden de convergencia y la constante asintótica se relacionan mediante

$$r = -\log \lambda$$

El orden de convergencia indica que "multiplica por p el número de cifras exactas de la solución" (Cisneros, 2017, p.16).

Cisneros (2017, p.18) manifiesta que existe una correspondencia directa entre la cantidad de iteraciones de un esquema y la eficiencia computacional:

Definición 15 *Si r es el orden de convergencia del método iterativo y d es el número de evaluaciones funcionales por iteración de dicho método, entonces, la eficiencia computacional está dada por*

$$\rho = r^{1/d} = \sqrt[d]{r}$$

Siempre se busca que un esquema tenga una mayor eficiencia computacional que los demás esquemas.

Sharma (2003, p.710) señala un método para encontrar el orden de convergencia computacional, si se han encontrado tres iteraciones.

Definición 16 *Suponga que x_{n+1} , x_n y x_{n-1} son tres iteraciones consecutivas cercanas a la raíz α. El orden de convergencia computacional ρ puede ser aproximado usando*

$$\rho = \frac{\ln\left|\dfrac{x_{n+1}-\alpha}{x_n-\alpha}\right|}{\ln\left|\dfrac{x_n-\alpha}{x_{n-1}-\alpha}\right|}$$

Criterios de Parada

En los métodos iterativos se debe elegir una condición para que el programa se detenga, y muestre una aproximación razonable, dada las múltiples ventajas, Cisneros (2017, p.20) demuestra que un buen criterio de parada es $\left|\frac{x_n-x_{n-1}}{x_n}\right| \leq \varepsilon$ por lo tanto este será usado como criterio de parada de todos los algoritmos presentados. Se puede observar que es equivalente a decir

$$e_r \leq \epsilon$$

Donde ϵ es un valor definido a criterio del análisis realizado.

Para el caso de los algoritmos bipaso, el error relativo puede encontrarse como

Definición 17 *Sean x_n, y_n dos iteraciones consecutivas de un algoritmo bipaso, el error relativo puede aproximarse mediante*

$$e_r = \left| \frac{x_n - y_n}{x_n} \right|$$

7.2. Método del Punto Fijo

Los puntos fijos permiten aproximar raíces mediante una transformación en la ecuación, por ejemplo $f(x) = 0$ mediante una transformación algebraica en la forma $x = g(x)$ es capaz de generar soluciones a la ecuación original.

Burden et al (2017, p. 41) definen punto fijo de la siguiente manera

Definición 18 *Un número x es un punto fijo para una función dada si $g(x) = x$.*

Una propiedad que permite el uso de los puntos fijos para crear métodos iterativos, según Arébalo, Bernal y Posada (2017, p.21) es:

Teorema 1 *Sean $[a, b] \subseteq \mathbb{R}$ y $g : [a, b] \to [a, b]$ es una función continua. Entonces g tiene un punto fijo en $[a, b]$.*

Demostración 1 *Si $g(a) = a$ o $g(b) = b$, entonces un punto fijo es un extremo, en caso contrario, se tiene $g(a) > a$, y $g(b) < b$. Dado que $f(x) = g(x) - x$, es una función continua en $[a, b]$ y $f(a) = g(a) - a > 0$ y $f(b) = g(b) - b < 0$, por el teorema del valor intermedio se sigue que f tiene un cero en $[a, b]$, es decir, existe $p \in [a, b]$ tal que $f(p) = 0$, lo que implica que $g(p) - p = 0$, o $g(p) = p$.*$\blacksquare$

Otra característica de los puntos fijos citada por Burden et al (2017, p.42), es expresada en el teorema adjunto:

Teorema 2 *El punto fijo es único.*

Burden et al (2017, p.42) muestran una prueba de este teorema, similar a la siguiente:

Demostración 2 *Supongase que existen dos puntos fijos p y q, según el teorema del valor medio, deben cumplir que existe ξ de forma que $p < \xi < q$ y que cumple*

$$\frac{g(p) - g(q)}{p - q} = g'(\xi)$$

Si $|g'(\xi)| \leq k \leq 1$, entonces

$$|p-q| = |g'(\xi)|\,|p-q| \leq k|p-q| < |p-q|$$

Por lo tanto, el punto fijo es único.

A partir del método del punto fijo se puede crear un esquema iterativo, como lo señala Cisneros (2017, p.22):

Sea $\{x_n\}$ la sucesión generada a partir de $x_0 \in [a,b]$ mediante la iteración $x_{n+1} = g\left(x_n\right)$, $n \geq 0$ y $L = \max\limits_{x\in[a,b]} |g'\left(x\right)| < 1$. Se verifica entonces que

$$e_n = |x_n - p| = |g\left(x_{n-1}\right) - g\left(p\right)| = |g'\left(x\right)|\,|e_{n-1}| \leq Le_{n-1} \leq L^2 e_{n-2} \leq \cdots \leq L^n e_0 \qquad \blacksquare$$

Teorema 3 *Sea $g : I \subset [a;b] \rightarrow R$ una función continua en $[a,b]$. Si $\{x_n\}$ es una sucesión en $[a,b]$ que converge a $\alpha \in [a,b]$, entonces α es un punto fijo de g.*

Demostración 3 *Como $\lim\limits_{n\to\infty}\left(x_n\right) = \alpha$, se tiene que $\lim\limits_{n\to\infty}\left(x_{n+1}\right) = \alpha$. A su vez, como se tiene que, g es continua en $[a,b]$ y por la relación $x_{n+1} = g(x_n)$, se obtiene*

$$g\left(\alpha\right) = g\left(\lim_{n\to\infty}\left(x_n\right)\right) = \lim_{n\to\infty} g\left(x_n\right) = \lim_{n\to\infty} x_{n+1} = \alpha$$

de donde α es el punto fijo de g. $\blacksquare$

Estos teoremas permiten establecer las características de los puntos fijos:

1. Se genera una sucesión convergente a los puntos fijos.
2. Los puntos fijos son únicos.

Por lo tanto, a partir de un punto fijo, se puede generar un esquema iterativo donde el valor semilla es $x_0 \in [a,b]$ y si $n = 1, 2, \cdots$, entonces el siguiente término de la sucesión viene dado por $x_{n+1} = g(x_n)$

7.3. Métodos Iterativos

Un método iterativo para resolver ecuaciones usa los resultados anteriores para obtener algunas aproximaciones más exactas a la solución de una ecuación. Se trata de una sucesión convergente, donde cada término se obtiene mediante una iteración específica, por lo tanto, los algoritmos son sucesiones definidas recurrentemente. Según Samarski (1988), un método iterativo “permite hallar la solución aproximada del sistema, construyendo una sucesión de aproximaciones (iteraciones), a partir de cierta aproximación inicial" (p.105). A la aproximación inicial, se le llama valor semilla, en el caso de las ecuaciones no lineales, se elige un valor cercano a la solución exacta, luego el algoritmo genera aproximaciones, idealmente cada aproximación debe acercarse al valor de la raíz real, es decir disminuir la diferencia entre la raíz y la aproximación nueva.

Teoría General de los Métodos Iterativos

Sea la serie $x_n, x_{n-1},\ x_{n-2},\ \cdots, x_{n-m+1}$, donde $n = m,\ m+1,\ \cdots$ y además $n - m + 1 > 0$. Un método iterativo se define como:

$$x_{n+1} = g\left(x_n, x_{n-1},\ x_{n-2},\ \cdots, x_{n-m+1}\right)\ ,\ n = m,\ m+1,\ \cdots$$

La función g se llama función de iteración, y $\{x_n\}_{n=0}^{\infty}$ la sucesión de iterados.

Para simplificar la teoría solo se consideran métodos iterativos donde $m = 1$. Es decir donde el método ignora todos los términos de la sucesión excepto el último (n) para general el siguiente $(n+1)$, entonces la definición de método numérico para este caso en específico se convierte en:

$$x_{n+1} = g(x_n),\ n = 0, 1, 2, ...$$

Un ejemplo de un método iterativo donde $m = 1$ es el método de Newton

$$g\left(x\right) = x - \frac{f\left(x\right)}{f'\left(x\right)}$$

Sea $\{x_n\}_{n=0}^{\infty}$ una sucesión con un primer término x_0 inicial, además $x_n \to \alpha$ y g es una función continua. Si

$$\alpha = g\left(\alpha\right)$$

α es un punto fijo de g.

Según Burden et al (2017, p. 45), un método iterativo de punto fijo puede ser codificado de la siguiente manera:

ENTRADA Aproximación inicial p_0; tolerancia TOL; número de iteraciones N_0

SALIDA aproximada p o mensaje de falla.

Paso 1 Determine $i = 1$.

Paso 2 Mientras $i \leq N_0$ haga los pasos 3-6.

Paso 3 Determine $p = g(p_{0)}.$

Paso 4 Si $|p - p_0| < TOL$ entonces

SALIDA (p) (El procedimiento fue exitoso.)

PARE

Paso 5 Determine $i = i + 1$

Paso 6 Determine $p_0 = p$.(Actualizar p_0)

Paso 7 SALIDA (El procedimiento falló después de N_0 iteraciones.)

PARE.

Este el algoritmo general, aplicado en un programa computacional para encontrar un punto fijo cualquiera.

7.4. Método de Newton

Se sabe que el esquema iterativo del método de Newton está definido por

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

donde la función $f(x)$ es diferenciable, monótona y además $f'(x) \neq 0$.

Una forma de demostrar el método de Newton es mediante los polinomios de Taylor, de la misma forma que usó Burden (2017, p.50) y Cisneros (2017, p.36). Si se busca encontrar el valor α el cual es raíz de la ecuación $f(x) = 0$ y se toma como primera aproximación o valor semilla el valor x_0 se obtiene

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + O\left(h^2\right)$$

Si $O(h^2) \approx 0$ y por lo tanto despreciable, luego se iguala a cero,

$$\begin{aligned}
0 &= f(x_0) + f'(x_0)\,x - f'(x_0)\,x_0 \\
f'(\alpha)\,x &= f'(x_0)\,x_0 - f(x_0) \\
x &= \frac{f'(x_0)\,x_0 - f(x_0)}{f'(x_0)} \\
x &= \frac{f'(x_0)\,x_0}{f'(x_0)} - \frac{f(x_0)}{f'(x_0)} \\
x &= x_0 - \frac{f(x_0)}{f'(x_0)}
\end{aligned}$$

Por lo tanto, se forma un esquema iterativo,

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

Con una función $f(x)$ que es continua, monótona, derivable en el intervalo donde se encuentra la raíz α y $f'(\alpha) \neq 0$.

7.5. Análisis de Convergencia del Método de Newton

A partir del método de Newton

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

Si se le cambian las variables

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

y se hace una transformación algebraica

$$h'(x_n)(x_{k+1} - x_k) = -h(x_k)$$

se obtiene que,

$$h(x_k) + h'(x_k)(x_{k+1} - x_k) = 0, \ \forall\ k \geq 0$$

El polinomio de Taylor está expresado como

$$h(x) = \Sigma_{i=0}^{n} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

por lo tanto

$$h(\alpha_k) = h(x_k) + h'(x_k)(\alpha_k - x_k) + \frac{1}{2}h''(x_k)(\alpha_k - x_k)^2 \quad , \quad \forall\ k \geq 0$$

donde α_k es un punto situado entre α y x_k.

$$h(\alpha_k) - h(x_k) = h'(x_k)(\alpha_k - x_k) + \frac{1}{2}h''(x_k)(\alpha_k - x_k)^2 \quad , \quad \forall\ k \geq 0$$

Y si se eliminan los términos semejantes resulta,

$$h'(x_k)(\alpha_k - x_k) + \frac{1}{2}h''(\alpha_k)(\alpha_k - x_k)^2 = 0 \quad , \quad \forall\ k \geq 0$$

Despejando $(\alpha_k - x_k)$ se tiene que,

$$h'(x_k)(\alpha_k - x_k) = -\frac{1}{2}h''(\alpha_k)(\alpha_k - x_k)^2 \quad , \quad \forall\ k \geq 0$$

$$(\alpha_k - x_k) = -\frac{1}{2}\frac{h''(\alpha_k)}{h'(x_k)}(\alpha_k - x_k)^2 \quad , \quad \forall\ k \geq 0$$

además:

$$|x_k - \alpha_k| = \left| \frac{1}{2}\frac{h''(\alpha_k)}{h'(x_k)}(x_k - \alpha_k)^2 \right| \quad , \quad \forall\ k \geq 0$$

si

$$p = \min_{x \in [a,b]} |f'(x)| \quad y \quad q = \max_{x \in [a,b]} |f''(x)|$$

de la expresión anterior obtenemos

$$|x_k - \alpha| \leq \frac{q}{2p}|x_k - \alpha|^2 \quad , \quad \forall\ k \geq 0$$

El orden de convergencia de Newton es cuadrático. ■

7.6. Polinomios de Adomian

Los polinomios de Adomian se construyeron para resolver ecuaciones diferenciales, pero tienen otros usos como ayuda al diseño de esquemas iterativos. Karthikeyan (2016), señala que, desde los años ochenta, el método de descomposición de Adomian, ha sido aplicado a diversas clases de ecuaciones funciones y para diversos fines en por ejemplo .ecuaciones lineales y no lineales, deterministas o estocásticas"(Lizarrága y Shingareva, 2012, p.21). Karthikeyan (2016) también manifiesta la naturaleza en forma de series infinitas generalmente convergentes que se obtienen mediante la aplicación de la descomposición de Adomian.

Para los fines de este trabajo se considera la ecuación no lineal $f(x) = 0$, $\alpha \in \mathbb{R}$ con una raíz de multiplicidad 1 de f. Según Cisneros (2017, p.52) y Basto, Semiao y Calheiros (2005, p.471), el método consiste en construir una función en la forma

$$h = c + N(h)$$

donde c es una constante y N es una función no lineal.

Basto et al (2005, p.469), mencionan que la solución de la ecuación tiene la forma

$$h = \sum_{n=0}^{\infty} h_n$$

y la función no lineal N se descompone como

$$N(h) = \sum_{n=0}^{\infty} A_n$$

Según Basto, et al (2005, p.470) a los términos A_n se les llama polinomios de Adomian y según Cisneros (2017, p.52), están dados por la siguiente serie general

$$A_n = \frac{1}{n!}\frac{d^n}{d\lambda^n}\left[N\left(\sum_{i=0}^{\infty}\lambda^i h_i\right)\right]_{\lambda=0} \quad , \quad n = 0, 1, 2, 3, \cdots$$

Los primeros tres polinomios de Adomian son

$$\begin{aligned} A_0 &= N\left(\sum_{i=0}^{\infty}\lambda^i h_i\right)_{\lambda=0} \\ A_0 &= N(h_0) \end{aligned}$$

El primer polinomio de Adomian, Cisneros (2017, p.52), lo expresa de la siguiente forma

$$\begin{aligned} A_1 &= \frac{d}{d\lambda}\left[N\left(\sum_{i=0}^{\infty}\lambda^i h_i\right)\right]_{\lambda=0} \\ A_1 &= N'\left(\sum_{i=0}^{\infty}\lambda^i h_i\right)\left(\sum_{i=0}^{\infty}i\lambda^{i-1}h_i\right) = h_1 N'(h_0) \end{aligned}$$

Según Cisneros (2017, p.52) y Karthikeyan (2016, p. 4), el segundo polinomio de Adomian es

$$\begin{aligned}
A_2 &= \frac{1}{2}\frac{d^2}{d\lambda^2}\left[N\left(\sum_{i=0}^{\infty}\lambda^i h_i\right)\right]_{\lambda=0} \\
A_2 &= \frac{1}{2}\left[N''\left(\sum_{i=0}^{\infty}\lambda^i h_i\right)\left(\sum_{i=0}^{\infty}i\lambda^{i-1}h_i\right)^2 + N'\left(\sum_{i=0}^{\infty}\lambda^i h_i\right)\left(\sum_{i=0}^{\infty}i\left(i-1\right)\lambda^{i-2}h_i\right)\right]_{\lambda=0} \\
A_2 &= \frac{1}{2}\left[h_1^2 N''\left(h_0\right) + 2h_2 N'\left(h_0\right)\right] \\
A_2 &= \frac{1}{2}h_1^2 N''\left(h_0\right) + h_2 N'\left(h_0\right)
\end{aligned}$$

Por definición los polinomios de Adomian son convergentes a la función original y pueden usarse para obtener aproximaciones numéricas de las ecuaciones no lineal a considerar o esquemas similares.

7.7. Construcción del Método de Newton por el método de Adomian

El método de Newton se puede construir a partir de los polinomios de Adomian. Cisneros (2017, p.53) usa una demostración similar a la siguiente para probar el método de Newton desde los polinomios de Adomian:

Si se tiene un intervalo I del conjunto de los números reales, $I \subset \mathbb{R}$, y además $f : I \to \mathbb{R}$ una función continua, monótona y diferenciable en el intervalo I de clase C^3, $f\left(x\right) = 0$ una ecuación no lineal y α una raíz de f que se encuentra en el intervalo I.

El Teorema de Taylor a $f\left(x-h\right)$ alrededor de x, se tiene

$$f\left(x-h\right) = f\left(x\right) + f'\left(x\right)\left(x-h-x\right) + \frac{f''\left(x\right)}{2!}\left(x-h-x\right)^2 + O\left(h^3\right)$$

si se simplifica se obtiene,

$$f\left(x-h\right) = f\left(x\right) - hf'\left(x\right) + h^2\frac{f''\left(x\right)}{2!} + O\left(h^3\right)$$

Donde $O\left(h^3\right)$ es el error, para una h suficientemente pequeño, se tiene que

$$f\left(x-h\right) = 0 \approx f\left(x\right) - hf'\left(x\right)$$

de donde

$$h = \frac{f\left(x\right)}{f'\left(x\right)}$$

haciendo

$$h = c + N(h)$$

donde

$$c = \frac{f(x)}{f'(x)} \quad y \quad N(h) = 0$$

para $n = 0$, obtenemos de $h = \sum_{n=0}^{\infty} h_n$ que

$$h \approx h_0 = c$$

como $f(x - h) = 0$, se tiene

$$\alpha = x - h \approx x - h_0 = x - c = x - \frac{f(x)}{f'(x)}$$

finalmente, la forma recurrente del método de Newton se obtiene cambiando la variable independiente por x_n.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad , \quad n \geq 0$$

7.8. Método iterativo Abbs

Un método reciente para encontrar raíces es el método iterativo de Abbasbandy fue demostrado por Saeid Abbasbandy mediante la técnica de descomposición de los polinomios de Adomian. Según Cisneros (2017, p.54) tiene una convergencia cúbica y la eficiencia computacional del método es $3^{1/3} \approx 1{,}44225$

Construcción del Método

La demostración siguiente es similar a la propuesta realizada por Cisneros (2017, p.54).

Si $I \subset \mathbb{R}$ un intervalo, $f : I \to \mathbb{R}$ una función de clase C^3 en I y $\alpha \in I$ la cual es continua, monótona, si esta función se iguala a cero se obtiene la ecuación no lineal $f(x) = 0$. Aplicando el teorema de Taylor de segundo grado es $f(x - h)$ alrededor de x, obtenemos

$$f(x - h) = f(x) + f'(x)(x - h - x) + \frac{f''(x)}{2!}(x - h - x)^2 + O\left(h^3\right)$$

Donde $O(h^3)$ es el error. Si h es cercano a cero entonces se obtiene que

$$\begin{aligned} f(x - h) &\approx f(x) - hf'(x) + \frac{h^2}{2} f''(x) \\ 0 &= f(x) - hf'(x) + \frac{h^2}{2} f''(x) \end{aligned}$$

$$
\begin{aligned}
hf'(x) &= f(x) + \frac{h^2}{2} f''(x) \\
h\frac{f'(x)}{f'(x)} &= \frac{f(x)}{f'(x)} + \frac{h^2}{2}\frac{f''(x)}{f'(x)} \\
h &= \frac{f(x)}{f'(x)} + \frac{h^2}{2}\frac{f''(x)}{f'(x)} \\
h &= h_0 + h_1
\end{aligned}
$$

tomando

$$c = h_0 = \frac{f(x)}{f'(x)} \quad y \quad h_1 = N(h) = \frac{h^2}{2}\frac{f''(x)}{f'(x)}$$

resulta que

$$h = c + N(h)$$

aplicando el método de descomposición de Adomian a h, para $n = 2$ se tiene que

$$h \approx h_0 + h_1 + h_2$$

entonces

$$h_0 = c \quad , \quad h_1 = A_0 \quad , \quad h_2 = A_1$$

donde

$$
\begin{aligned}
h_0 &= \frac{f(x)}{f'(x)} \\
h_1 &= N(h_0) = \frac{h_0^2}{2}\frac{f''(x)}{f'(x)} = \frac{f^2(x)\, f''(x)}{2\left[f'(x)\right]^3} \\
h_2 &= h_1 N'(h_0) = h_1 h_0 \frac{f''(x)}{f'(x)} = \frac{f^3(x)\left[f''(x)\right]^2}{2\left[f'(x)\right]^5}
\end{aligned}
$$

se tiene que

$$\alpha = x - h \approx x - \frac{f(x)}{f'(x)} - \frac{f^2(x)\, f''(x)}{2\left[f'(x)\right]^3} - \frac{f^3(x)\left[f''(x)\right]^2}{2\left[f'(x)\right]^5}$$

Por lo tanto, se concluye que el método de Abbasbandy está expresado por:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)\, f''(x_n)}{2\left[f'(x_n)\right]^3} - \frac{f^3(x_n)\left[f''(x_n)\right]^2}{2\left[f'(x_n)\right]^5}, \; n \geq 0$$

7.9. Métodos iterativos de Chun

Los métodos iterativos de Chun son métodos multipasos para determinar la raíz de las ecuaciones no lineales. Básicamente el método consiste en encontrar dos sucesiones una de $\{x_n\}$ y otra de $\{y_n\}$. Por ejemplo, Cisneros (2017) dice que el "método presenta una convergencia de orden cuatro. Además, la eficiencia computacional del método es: $4^{1/4} \approx 1,41421$"(p.55).

Citando a Cisneros (2017, p.55) el esquema iterativo del método de Chun está dado por

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
x_{n+1} &= y_n - 2\frac{f(y_n)}{f'(y_n)} + \frac{f(y_n)\,f'(y_n)}{[f'(x_n)]^2},\ n \geq 0
\end{aligned}
$$

Chun (2005, p.1564) logró modificar el método de Newton empleando polinomios de Adomian, obteniendo algoritmos bipasos, como:

$$
\begin{aligned}
x_{n+1}^* &= x_n - \frac{f(x_n)}{f'(x_n)} \\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} - 3\frac{f\left(x_{n+1}^*\right)}{f'(x_n)} + 3\frac{f(x_{n+1}^*)f'(x_{n+1}^*)}{f(x_n)^2} \\
&\quad -\frac{1}{2}\frac{f(x_{n+1}^*)(f(x_{n+1}^*)f''(x_{n+1}^*) + 2(f'(x_{n+1}^*))^2}{[f'(x_n)]^3}
\end{aligned}
$$

7.10. Método de variación iteracional

Según Ganji et al (2006, p.180), si se presenta la ecuación diferencial $Lu + Nu = g(x)$,donde L es un operador lineal y N es un operador no lineal y $g(x)$ es un término heterogéneo, se puede construir una función como sigue

$$u_{n+1}(x) = u_n(x) + \int_0^x \lambda \left\{Lu_n(\tau) + N\widetilde{u}_n(\tau) - g(\tau)\right\} d\tau$$

Donde λ es un multiplicador de Lagrange que puede ser optimizado por la iteración variacional.

7.11. Programación de Algoritmos en Python

Un programa "no es nada más que una serie de instrucciones dadas al ordenador en un lenguaje entendido por él, para decirle exactamente lo que queremos que haga" (Ceballos, 2007, p.2). Se ingresan datos al ordenador, el computador lee y realiza una acción, permitiendo automatizar, facilitar y simplificar las actividades al usuario. Actualmente la sociedad está adaptada a los programas computacionales, cada uno de ellos posee un fin específico, y un uso particular, dependiendo de la necesidad del usuario. Un programa “se escribe en un lenguaje de programación” ((Johanes, Fernández, y Rodriguez, 2003, p.2), de aquí se concluye que la importancia de los lenguajes de programación es vital, porque permiten crear programas nuevos.

Johanes et al (2003, p.3), señala que existen tres tipos de lenguajes de programación, lenguaje de máquina, lenguaje de bajo nivel y lenguaje de alto nivel. Los lenguajes de alto nivel “proporcionan sentencias muy fáciles de recordar, que no dependen del tipo de computadora y han de traducirse a lenguaje máquina por unos programas denominados compiladores o intérpretes” (Johanes et al, 2003, p.4), las ventajas de este tipo de lenguaje les permite ser elegidos por un público amplio, y los lenguajes de alto nivel tienen preferencia sobre los otros tipos, en especial al momento de crear programas.

En la actualidad existen diferentes lenguajes de programación con diferentes criterios para ser usados, algunos de ellos “son muy populares, como Java y C++. Otros son menos conocidos, como Squeak y T. Algunos otros están diseñados para facilitar en gran medida el aprendizaje de las ideas de ciencias computacionales, como Scheme o Python” (Guzdial y Ericson, 2013, p.6). Debido a la facilidad que tiene de ser aprendido que posee Python, es preferible el uso de Python sobre otros lenguajes. También se pueden señalar las siguientes virtudes: “su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo” (González, 2015, p.7). La elección de Python para copilar los códigos radica en su sencillez, a la existencia de muchas librerías relacionadas con matemáticas, además “ofrece mucho más chequeo de error que C” (Rossun, 2017, p.11) y a la facilidad del aprendizaje del programa.

El siguiente concepto importante a examinar es el de algoritmo, un algoritmo es “es la descripción de un proceso paso a paso que no está enlazado con ningún lenguaje de programación”, (Guzdial y Ericson, 2013, p.8), manifestando que los algoritmos son un conjunto de pasos, ordenados y finitos, que no requieren de un lenguaje un lenguaje de programación. Para expresar un algoritmo en un esquema básico de programación se puede usar un pseudocódigo, esto es “la representación textual de un algoritmo de manera que dicho texto se encuentre enmarcado en algunas normas técnicas que faciliten su posterior transcripción a un lenguaje de programación” (Trejos, 1999, p. 66), es decir, es la descripción de un algoritmo, para realizar pseudocódigos es conveniente el uso de PSint.

Un objeto en Python es "una forma ordenada de agrupar datos" (Curia, Manterola, Medrano, Páez y Wachenchauzer, 2011, p.141), puede entenderse un objeto como un nombre (sustantivo) que se comporta de cierta forma determinada a criterio del usuario a esta forma se le llama clase, es decir una clase "indica cuáles son los atributos y métodos que van a tener todas las variables que sean de ese tipo" (Curia et al, 2011, p.141), las clases determinan las propiedades y el comportamiento de los objetos. En Python, las clases se codifican como *class*, las cuales requieren de la entrada _ _*init*_ _ funcionando como el nombre del objeto.

El análisis orientado a objetos es "proceso de identificar los objetos, lo que cada uno de ellos sabe (respecto al problema)" (Guzdial y Ericson, 2013, p.377), y la programación orientada a objetos implica que se "se definen variables (variables de instancia) y funciones (métodos) para los objetos" (Guzdial y Ericson, 2013, p.377). La programación orientada a objetos permite trabajar con sencillos problemas que bajo otro paradigma serían complicados.

Según Rossum (2017, p.73), Python cuenta con colecciones de funciones relacionadas con Matemática en los módulos *math*, *SciPy,* también se encuentran algunas en la librería *numpy* para invocar funciones de cada módulo se escribe el nombre del módulo y luego para invocar las funciones una expresión inicial como *math.* o *np.* a cada función.

Un módulo en Python "es un archivo de Python que contiene variables, funciones y clases" (Vadillo, 2019, p.41), a veces para acceder a una de estas funciones puede "usarse la palabra reservada *import…from"* (Vadillo, 2019, p.41).

Una función en Python es "un fragmento de código con un nombre asociado que realiza una serie de tareas y devuelve un valor" (González, 2015, p.35). Para nombrar las funciones se "escribe la palabra reservada def seguida del nombre de la función y sus parámetros entre paréntesis" (Vadillo, 2019, p.28).

8. CAPÍTULO IV: VARIANTES DE ADOMIAN

El objetivo central de este capítulo es considerar diferentes variantes de la función $N(h)$ y mostrar que se pueden obtener diferentes esquemas iterativos y a partir de aquí, nuevos métodos iterativos, todos ellos basado en la técnica de descomposición de Adomian y la combinación de las Técnicas Iterativas Variacional.

Dado que el método de Newton solo depende de una variable

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

Se puede considerar la siguiente notación $f(x_n) = f$ y $f'(x_n) = f'$. Por lo tanto el método de Newton está dado por

$$x_{n+1} = x_n - \frac{f}{f'}$$

Que se puede expresar como

$$x_{n+1} = x_n - A_0$$

Donde A_0 sería el primer polinomio de Adomian

$$A_0 = N(h_0)$$

$$A_0 = \frac{f}{f'}$$

y, además

$$h_0 = \frac{f}{f'}$$

Si se deriva A_1 para obtener el segundo término de los polinomios de Adomian.

$$\begin{aligned}
A_1 &= \frac{d}{d\lambda} N(h_0 + h_1\lambda)|_{\lambda=0} = h_1 N'\left(h_0 + h_1(0)\right) = h_1 N'(h_0) \\
A_1 &= h_1 N'(h_0) \\
h_1 &= \frac{f^2 \cdot f''}{2f'^3} \\
N'(h_0) &= \frac{d}{dh_0} \frac{h_0^2}{2} \frac{f''}{f'} = \frac{2h_0}{2} \frac{f''}{f'} = h_0 \frac{f''}{f'} = \left(\frac{f}{f'}\right)\left(\frac{f''}{f'}\right) = \frac{f \cdot f''}{f'^2}
\end{aligned}$$

Por lo tanto

$$A_2 = h_1 N'(h_0) = \left(\frac{f^2 f''}{2f'^3}\right)\left(\frac{f f''}{f'^2}\right) = \frac{f^3 f''^2}{2f'^5} = h_2$$

Nuevamente, para obtener el tercer coeficiente

$$
\begin{aligned}
A_3 &= \frac{1}{2}\frac{d}{d\lambda}(h_1+2h_2\lambda)N'(h_0+h_1\lambda)|_{\lambda=0} \\
A_3 &= h_2N'(h_0+h_1\lambda)|_{\lambda=0}+\frac{1}{2}h_1(h_1+2h_2\lambda)N'(h_0+h_1\lambda)|_{\lambda=0} \\
A_3 &= h_2N'(h_0+h_1\lambda)|_{\lambda=0}+\frac{1}{2}h_1(h_1+2h_2\lambda)N''(h_0+h_1\lambda)|_{\lambda=0} \\
A_3 &= h_2N'(h_0)+\frac{1}{2}h_1h_1N''(h_0) \\
A_3 &= h_2N'(h_0)+\frac{1}{2}h_1^2N''(h_0)
\end{aligned}
$$

Considerando que

$$h_0=\frac{f}{f'};\ h_1=\frac{f^2\cdot f''}{2f'^3};\ h_2=\frac{f^3\cdot f''^2}{2f'^5}$$

Y, además

$$
\begin{aligned}
N'(h_0) &= \frac{f\cdot f''}{f'^2} \\
N''(h_0) &= \frac{d}{dh_0}\left(h_0\frac{f''}{f'}\right)=\frac{f''}{f'}
\end{aligned}
$$

Por lo tanto, el tercer término de Adomian se obtiene mediante el siguiente razonamiento

$$
\begin{aligned}
A_3 &= h_2N'(h_0)+\frac{1}{2}h_1^2N''(h_0) \\
A_3 &= \left(\frac{f^3\cdot f''^2}{2f'^5}\right)\left(\frac{f\cdot f''}{f'^2}\right)+\frac{1}{2}\left(\frac{f^2\cdot f''}{2f'^3}\right)^2\left(\frac{f''}{f'}\right) \\
A_3 &= \frac{f^4\cdot f''^3}{2f'^7}+\frac{1}{2}\left(\frac{f^4\cdot f''^2}{4f'^6}\right)\frac{f''}{f'} \\
A_3 &= \frac{f^4\cdot f''^3}{2f'^7}+\frac{f^4\cdot f''^3}{8f7}=\frac{5}{8}\frac{f^4\cdot f''^3}{f'^7} \\
A_3 &= h_3=\frac{5}{8}\frac{f^4\cdot f''^3}{f'^7}
\end{aligned}
$$

Para el cuarto término, se sabe que

$$A_2 = h_1 N'(h_0)$$

$$A_3 = h_2 N'(h_0) + \frac{1}{2} h_1^2 N''(h_0)$$

luego

$$A_4 = \frac{1}{3}\frac{d}{d\lambda}((h_2 + 3h_3\lambda)N(h_0 + h_1\lambda) + \frac{1}{2}(h_1 + 2h_2\lambda)^2 N''(h_0 + h_1\lambda))|_{\lambda=0}$$

$$\begin{aligned}
A_4 &= \frac{1}{3}\left[3h_3 N'(h_0 + h_1\lambda) + h_1(h_2 + 3h_3\lambda)N''(h_0 + h_1\lambda)\right]|_{\lambda=0} + \\
&\frac{1}{3}\left(\frac{1}{2}\right)\left[2(2h_2)(h_1 + 2h_2\lambda)N''(h_0 + h_1\lambda) + h_1(h_1 + 2h_2\lambda)^2 N'''(h_0 + h_1\lambda)\right]|_{\lambda=0} \\
A_4 &= \frac{1}{3}\left[3h_3 N'(h_0) + h_1(h_2)N''(h_0)\right] + \left(\frac{1}{3}\right)\left(\frac{1}{2}\right)\left[(4h_2)(h_1)N''(h_0) + h_1(h_1^2)N''(h_0)\right] \\
A_4 &= \frac{1}{3}\left[3h_3 N'(h_0) + h_1 h_2 N''(h_0)\right] + \frac{1}{6}\left[2h_1 h_2 N''(h_0) + h_1^3 N'''(h_0)\right] \\
A_4 &= \frac{1}{6}\left[6h_3 N'(h_0) + 2h_1 h_2 N''(h_0)\right] + \frac{1}{6}\left[4h_1 h_2 N''(h_0) + h_1^3 N'''(h_0)\right] \\
A_4 &= \frac{1}{6}\left[6h_3 N'(h_0) + 6h_1 h_2 N''(h_0) + h_1^3 N'''(h_0)\right] \\
A_4 &= \frac{1}{6}\left[6h_3 N'(h_0) + 6h_1 h_2 N''(h_0) + h_1^3 N'''(h_0)\right]
\end{aligned}$$

Como

$$N'(h_0) = \frac{d}{dh_0}\left(\frac{h_0^2}{2}\frac{f''}{f'}\right) = \frac{2h_0}{2}\frac{f''}{f'} = h_0\frac{f''}{f'} = \frac{f}{f'}\frac{f''}{f'} = \frac{ff''}{f'^2}$$

Entonces

$$N''(h_0) = \frac{d}{dh_0}\left(h_0\frac{f''}{f'}\right) = \frac{f''}{f'}$$

Por lo tanto,

$$\begin{aligned}
N'(h_0) &= \frac{d}{dh_0}\left(\frac{h_0^2}{2}\frac{f''}{f'}\right) = \frac{f \cdot f''}{f'^2} \\
N''(h_0) &= \frac{d}{dh_0}\left(h_0\frac{f''}{f'}\right) = \frac{f''}{f'} \\
N'''(h_0) &= \frac{d}{dh_0}\left(\frac{d}{dh_0}\left(h_0\frac{f''}{f'}\right)\right) = \frac{d}{dh_0}\left(\frac{f''}{f'}\right) = 0
\end{aligned}$$

Si se tiene presente los valores de h obtenidos anteriormente

$$
\begin{aligned}
h_1 &= \frac{f^2 \cdot f''}{2f'^3} \\
h_2 &= \frac{f^3 \cdot f''^2}{2f'^5} \\
h_3 &= \frac{f^4 \cdot f''^3}{2f'^7} + \frac{f^4 \cdot f''^3}{8f'^7} = \frac{5}{8}\frac{f^4 f''^3}{f'^7}
\end{aligned}
$$

Al sustituir resulta

$$
\begin{aligned}
A_4 &= \frac{1}{6}\left[6h_3N'(h_0) + 6h_1h_2N''(h_0) + h_1^3N'''(h_0)\right] \\
A_4 &= \frac{1}{6}\left[6\left(\frac{5}{8}\right)\left(\frac{f^4 \cdot f''^3}{f'^7}\right)\left(\frac{f \cdot f''}{f'^2}\right) + 6\left(\frac{f^2 f''}{2f'^3}\right)\left(\frac{f^3 \cdot f''^2}{2f'^5}\right)\left(\frac{f''}{f'}\right) + \left(\frac{f^2 \cdot f''}{2f'^3}\right)^3(0)\right] \\
A_4 &= \frac{1}{6}\left[\frac{30}{8}\left(\frac{f^5 \cdot f''^4}{f'^9}\right) + \frac{6}{4}\left(\frac{f^5 \cdot f''^4}{f'^9}\right) + 0\right] = \frac{1}{6}\left(\frac{42}{8}\frac{f^5 \cdot f''^4}{f'^9}\right) = \frac{42}{48}\frac{f^5 \cdot f''^4}{f'^9} = \frac{7}{8}\frac{f^5 \cdot f''^4}{f'^9} \\
A_4 &= h_4 = \frac{7}{8}\frac{f^5 \cdot f''^4}{f'^9}
\end{aligned}
$$

Para encontrar A_5 se tiene que

$$
A_4 = \frac{1}{6}\left[6h_3N'(h_0) + 6h_1h_2N''(h_0) + h_1^3N''(h_0)\right]
$$

Por lo tanto

$$
A_5 = \frac{1}{4}\frac{d}{d\lambda}A_4|_{\lambda=0}
$$

$$
A_5 = \frac{1}{24}\frac{d}{d\lambda}\left[\begin{array}{c} 6(h_3 + 4h_4\lambda)N'(h_0 + h_1\lambda) + 6(h_1 + 2h_2\lambda)(h_2 + 3h_3\lambda)N''(h_0 + h_1\lambda) + \\ (h_1 + h_2\lambda)^3N''(h_0 + h_1\lambda) \end{array}\right]|_{\lambda=0}
$$

$$
A_5 = \frac{1}{24}\left[\begin{array}{c} 6\left[4h_4N'(h_0 + h_1\lambda) + h_1(h_3 + 4h_4\lambda)N''(h_0 + h_1\lambda)\right] + \\ 6\left[\begin{array}{c} 2h_2(h_2 + 3h_3\lambda)N''(h_0 + h_1\lambda) + 3h_3(h_1 + 2h_2\lambda)N''(h_0 + h_1\lambda) + \\ h_1(h_1 + 2h_2\lambda)(h_2 + 3h_3\lambda)N'''(h_0 + h_1\lambda) \end{array}\right] \\ + \left[3(2h_2)(h_1 + 2h_2\lambda)^2N'''(h_0 + h_1\lambda) + h_1(h_1 + 2h_2\lambda)^3N^{(4)}(h_0 + h_1\lambda)\right] \end{array}\right]|_{\lambda=0}
$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 6\left[4h_4N'(h_0)+h_1(h_3)N'(h_0)\right]+6\left[\begin{array}{c} 2h_2(h_2)N''(h_0)+3h_3(h_1)N''(h_0)+ \\ h_1(h_1)(h_2)N'''(h_0) \end{array}\right]+ \\ \left[6h_2(h_1)^2N'''(h_0)+h_1(h_1)^3N^{(4)}(h_0)\right] \end{array}\right]$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 6\left[4h_4N'(h_0)+4h_1(h_3)N''(h_0)\right]+6\left[2h_2(h_2)N''(h_0)+2h_1(h_1)(h_2)N'''(h_0)\right]+ \\ \left[h_1(h_1)^3N^{(4)}(h_0)\right] \end{array}\right]$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 24h_4N'(h_0)+24h_1(h_3)N''(h_0)+12h_2^2N''(h_0)+48h_1(h_1)(h_2)N'''(h_0) \\ +h_1(h_1)^3N^{(4)}(h_0) \end{array}\right]$$

$$A_5 = \frac{1}{24}\left[24h_4N'(h_0)+(24h_1h_3+12h_2^2)N''(h_0)+12h_1^2h_2N'''(h_0)+h_1^4N^{(4)}(h_0)\right]$$

Hasta el momento se tiene que

$$N'(h_0) = \frac{d}{dh_0}\left(\frac{h_0^2}{2}\frac{f''}{f'}\right)=\frac{f\cdot f''}{f'^2}$$

$$N''(h_0) = \frac{d}{dh_0}\left(h_0\frac{f''}{f'}\right)=\frac{f''}{f'}$$

$$N'''(h_0) = \frac{d}{dh_0}\left(\frac{d}{dh_0}\left(h_0\frac{f''}{f'}\right)\right)=\frac{d}{dh_0}\left(\frac{f''}{f'}\right)=0$$

$$N^{(4)}(h_0)=0$$

y los valores de h *son*

$$h_1 = \frac{f^2\cdot f''}{2f'^3}$$

$$h_2 = \frac{f^3\cdot f''^2}{2f'^5}$$

$$h_3 = \frac{f^4\cdot f''^3}{2f'^7}+\frac{f^4\cdot f''^3}{8f7}=\frac{5}{8}\frac{f^4\cdot f''^3}{f'^7}$$

$$h_4 = \frac{7}{8}\frac{f^5\cdot f''^4}{f'^9}$$

y

$$A_5 = \frac{1}{24}\left[24h_4N'(h_0) + (24h_1h_3 + 12h_2^2)N''(h_0) + 12h_1^2h_2N'''(h_0) + h_1^4N^{(4)}(h_0)\right]$$

Como

$$N'''(h_0) = N^{(4)}(h_0) = 0$$

Se simplifica A_5 como

$$A_5 = \frac{1}{24}\left[24h_4N'(h_0) + (24h_1h_3 + 12h_2^2)N''(h_0)\right]$$

$$A_5 = \frac{1}{24}\left[24\left(\frac{7}{8}\frac{f^5\cdot f''^4}{f'^9}\right)\left(\frac{f\cdot f''}{f'^2}\right) + \left(24\left(\frac{f^2\cdot f''}{2f'^3}\right)\left(\frac{5}{8}\frac{f^4\cdot f''^3}{f'^7}\right) + 12\left(\frac{f^3\cdot f''^2}{2f'^5}\right)^2\right)\left(\frac{f''}{f'}\right)\right]$$

$$A_5 = \frac{1}{24}\left[24\left(\frac{7}{8}\right)\left(\frac{f^6\cdot f''^5}{f'^{11}}\right) + \left(\frac{120}{16}\frac{f^6\cdot f''^4}{f'^{10}} + 12\left(\frac{f^6\cdot f''^4}{4f'^{10}}\right)\right)\left(\frac{f''}{f'}\right)\right]$$

$$A_5 = \frac{1}{24}\left[\frac{168}{8}\frac{f^6\cdot f''^5}{f'^{11}} + \left(\frac{60}{8}\frac{f^6\cdot f''^4}{f'^{10}} + 3\frac{f^6\cdot f''^4}{f'^{10}}\right)\left(\frac{f''}{f'}\right)\right]$$

$$A_5 = \frac{1}{24}\left[\frac{168}{8}\frac{f^6\cdot f''^5}{f'^{11}} + \left(\frac{84}{8}\frac{f^6\cdot f''^4}{2f'^{10}}\right)\left(\frac{f''}{f'}\right)\right]$$

$$A_5 = \frac{1}{24}\left[\frac{168}{8}\left(\frac{f^6\cdot f''^5}{f'^{11}}\right) + \frac{84}{8}\left(\frac{f^6\cdot f''^5}{2f'^{11}}\right)\right]$$

$$A_5 = \frac{1}{24}\left[\frac{252}{8}\left(\frac{f^6\cdot f''^5}{f'^{11}}\right)\right]$$

$$A_5 = h_5 = \frac{252}{192}\frac{f^6\cdot f''^5}{f'^{11}} = \frac{21}{16}\frac{f^6\cdot f''^5}{f'^{11}}$$

Como un análisis final se presenta el resumen de la discusión anterior, en forma de los polinomios de Adomian.

$$
\begin{aligned}
A_0 &= \frac{f}{f'} = \frac{f(x_n)}{f'(x_n)} \\
A_1 &= \frac{f^2 \cdot f''}{2f'^3} = \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3} \\
A_2 &= \frac{f^3 \cdot f''^2}{2f'^5} = \frac{(f(x_n))\,(f''(x_n))^2}{2(f'(x_n))^5} \\
A_3 &= \frac{5}{8}\frac{f^4 \cdot f''^3}{f'^7} = \frac{(f(x_n))^2\,(f''(x_n))^3}{2(f'(x_n))^7} \\
A_4 &= \frac{7}{8}\frac{f^5 \cdot f''^4}{f'^9} = \frac{7}{8}\frac{(f(x_n))^5 (f''(x_n))^4}{(f'(x_n))^9} \\
A_5 &= \frac{21}{16}\frac{f^6 \cdot f''^5}{f'^{11}} = \frac{21}{16}\frac{(f(x_n))^6 (f''(x_n))^5}{(f'(x_n))^{11}}
\end{aligned}
$$

Mediante los polinomios de Adomian se generaron los siguientes algoritmos, los cuales fueron presentados en diversos artículos por diferentes autores:

1. Método de Newton, considerando el primer término ($A_0 = \dfrac{f(x_n)}{f'(x_n)}$) de los polinomios de Adomian.

$$x_{n+1} = x_n - A_0$$

   Sustituyendo A_0 resulta

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

2. Método de House Holder

   considerando el segundo término ($A_1 = \dfrac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3}$) de Adomian.

$$x_{n+1} = x_n - A_0 - A_1$$

   Entonces se obtiene El método de House Holder

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

   Existen diversos análisis de este método, aunque su deducción no se basa precisamente en los polinomios de Adomian, se hace mención en el artículo de Bahgat y Hafiz (2014, p.87).

3. Método de Abbasbandy

   Si se añade el tercer término ($A_2 = \dfrac{(f(x_n))^3 (f''(x_n))^2}{2(f'(x_n))^5}$) de Adomian.

$$x_{n+1} = x_n - A_0 - A_1 - A_2$$

Entonces se logra demostrar el método de Abbasbandy

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

Los siguientes algoritmos se llamarán Cisneros-Méndez por el tutor y el autor de este trabajo seguidos de un número para indicar el orden.

Como es natural la función $f(x)$ a considerar debe ser continua, monótona y derivable, y con $f'(x) \neq 0$. Si la función $f''(x) = 0$ los métodos se reducen al método de Newton clásico.

Se considera la siguiente sucesión $x_{n+1} = x_n - A_0 - A_1 - A_2 - A_3$ para construir el primer algoritmo.

1. Método de Cisneros-Méndez 1 (Contempla el desarrollo para los cuatros primeros polinomios de Adomian), aquí $n = 0, 1, 2, ...$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3} - \frac{(f(x_n))^3 (f''(x_n))^2}{2(f'(x_n))^5} - \frac{5}{8}\frac{(f(x_n))^4 (f''(x_n))^3}{(f'(x_n))^7}$$

2. Método Cisneros-Méndez 2 (Contempla el desarrollo para los cinco primeros polinomios de Adomian) con $n = 0, 1, 2, ...$

   Es decir, la serie queda

$$x_{n+1} = x_n - A_0 - A_1 - A_2 - A_3 - A_4$$

   Y al sustituirse por sus respectivos valores

$$\begin{aligned} x_{n+1} \;&=\; x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3} - \frac{(f(x_n))^3 (f''(x_n))^2}{2(f'(x_n))^5} - \frac{5}{8}\frac{(f(x_n))^4 (f''(x_n))^3}{(f'(x_n))^7} \\ &\quad -\frac{7}{8}\frac{(f(x_n))^5 (f''(x_n))^4}{(f'(x_n))^9} \end{aligned}$$

3. Método Cisneros-Méndez 3 (Contempla el desarrollo para los seis primeros polinomios de Adomian)

   La serie

$$x_{n+1} = x_n - A_0 - A_1 - A_2 - A_3 - A_4$$

   Entonces si se sustituye resulta el siguiente esquema iterativo:

$$\begin{aligned} x_{n+1} \;&=\; x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3} - \frac{(f(x_n))^3 (f''(x_n))^2}{2(f'(x_n))^5} - \frac{5}{8}\frac{(f(x_n))^4 (f''(x_n))^3}{(f'(x_n))^7} \\ &\quad -\frac{7}{8}\frac{(f(x_n))^5 (f''(x_n))^4}{(f'(x_n))^9} - \frac{21}{16}\frac{(f(x_n))^6 (f''(x_n))^5}{(f'(x_n))^{11}} \end{aligned}$$

   Donde $n = 0, 1, 2, ...$

8.1. Técnica de Iteración Variacional y Polinomios de Adomian

Partiendo del siguiente esquema numérico, definido por los polinomios de Adomian (llamado método House Holder) se pueden obtener un conjunto de algoritmos que proporcionan las raíces en una cantidad menor de iteraciones que el método de Newton.

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

y tomando la función auxiliar $h(x_n) = f(x_n) \cdot g(x_n)$, es decir, $h = f \cdot g$, luego

$$h' = f'g + g'f$$

y

$$h'' = f''g + 2f'g' + g''f$$

Y por lo tanto si se sustituye resulta

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

De este esquema general y variando la función auxiliar g por familia de funciones exponenciales, se obtiene los siguientes nuevos algoritmos.

Es importante señalar que un proceso similar al anterior fue usado por Noor (2007, p. 129) y Cisneros (2017) de forma exitosa para generar nuevos algoritmos. Shah (2012, p. 55) usa un valor de α de 0, $0,5$ y 1. Los valores de α son elegidos entre el conjunto de los números reales, cuyo valor absoluto es menor que 1, $|\alpha| \leq 1$ y luego se selecionan los más óptimos mediante prueba y error, luego de ser programados los algoritmos.

Algoritmo 1 *Sea la función auxiliar $g(x) = e^{-\alpha x}$ entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{(f' - \alpha f)} - \frac{f^2\left(f'' - 2\alpha f' + \alpha^2 f\right)}{2\left(f' - \alpha f\right)^3} \tag{1}$$

El valor óptimo de α es $0,01$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$.

El denominador del esquema $f' - \alpha f \neq 0$ en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación. Cabe aclarar que se puede aplicar el método, aunque la función f no posea segunda derivada.

Algoritmo 2 *Sea la función auxiliar $g(x) = e^{-\alpha f(x)}$ entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{f' - \alpha f' \cdot f} - \frac{f^2[f'' - 2\alpha f'^2 - \alpha f'' \cdot f + \alpha^2 f'^2 \cdot f]}{2[f' - af' \cdot f]^3} \tag{2}$$

Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$ en la raíz c.

El valor óptimo de α es $0,001$. El denominador del esquema $f' - \alpha f' \cdot f \neq 0$ en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación. Cabe aclarar que se puede aplicar el método, aunque la función f no posea segunda derivada.

Algoritmo 3 *Sea la función auxiliar $g(x) = e^{-\alpha f^2}$ entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{f' - 2\alpha f^2 \cdot f'} - \frac{f^2(f'' - 4\alpha f \cdot f'^2 - 2\alpha \cdot f'^2 \cdot f - 2\alpha \cdot f^2 \cdot f'' + 4\alpha^2 \cdot f^3 \cdot f'^2)}{2(f' - 2\alpha f^2 \cdot f')^3} \tag{3}$$

El valor óptimo de α es $0,001$. Para este algoritmo la función debe ser monótona y derivable con $f'(c) \neq 0$ en la raíz c.

El denominador del esquema $f' - 2\alpha f^2 \cdot f' \neq 0$ en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación. Cabe aclarar que se puede aplicar el método, aunque la función f no posea segunda derivada.

Algoritmo 4 *Sea la función auxiliar $g(x) = e^{-\alpha f'}$ entonces el esquema iterativo toma la forma*

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

El valor óptimo de α es $0,1$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$ en la raíz c.

El denominador del esquema $f' - \alpha f'' \cdot f \neq 0$ en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación. No se puede aplicar el método si la función f no posee segunda derivada, porque se reduce al método de Newton.

Algoritmo 5 *Sea la función auxiliar $g(x) = e^{-\alpha \frac{f''}{f'}}$ entonces el esquema iterativo toma la forma*

$$\begin{aligned} x_{n+1} \; &= \; x_n - \frac{f}{\left(f' - \alpha\left(\dfrac{f' \cdot f''' - f''^2}{f'^2}\right) f\right)} - \frac{f^2\left[f'' - 2\alpha \cdot \left(\dfrac{f' \cdot f''' - f''^2}{f'}\right)\right]}{2\left[f' - \alpha\left(\dfrac{f' \cdot f''' - f''^2}{f'^2}\right) f\right]^3} \\ & - \frac{f^2 \left[\begin{array}{c} -\alpha\left(\dfrac{f'^2 \cdot f^{(4)} - 3 \cdot f' \cdot f'' \cdot f''' + 2 \cdot f''^3}{f'^3}\right) f \\ +\alpha^2\left(\dfrac{f' \cdot f''' - f''^2}{f'^2}\right)^2 f \end{array}\right]}{2\left[f' - \alpha\left(\dfrac{f' \cdot f''' - f''^2}{f'^2}\right) f\right]^3} \end{aligned} \tag{5}$$

El valor óptimo de α es $0,001$. Para aplicar este algoritmo se requieren las siguientes condiciones: La función debe ser monótona, derivable y con $f'(c) \neq 0$ en la raíz c.

El denominador del esquema

$$f' - \alpha\left(\frac{f' \cdot f''' - f''^2}{f'^2}\right) f \neq 0$$

en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación. No se puede aplicar el método si la función f no posee segunda derivada, y tiene que tener derivadas hasta de cuarto orden.

Algoritmo 6 *Sea la función auxiliar $g(x) = e^{-\alpha f''}$ entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{(f' - \alpha f''' \cdot f)} - \frac{f^2\left[f'' - 2\alpha f''' \cdot f' + \left(-\alpha f^{(4)} + \alpha^2 f'''\right) f\right]}{2\left(f' - \alpha f''' \cdot f\right)^3} \tag{6}$$

El valor óptimo de α es $0,01$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$ en la raíz c.

El denominador del esquema $f' - \alpha f''' \cdot f \neq 0$ en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación. No se puede aplicar el método si la función f no posee segunda derivada, porque se reduce al método de Newton.

Algoritmo 7 *Sea la función auxiliar $g(x) = e^{-\alpha f''^2}$ entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{(f' - 2\alpha f'' \cdot f''' \cdot f)} - \frac{f^2\left[f'' - 4\alpha f' \cdot f'' \cdot f''' + \left(-2\alpha f''' - 2\alpha f'' \cdot f^{(4)} + 4\alpha^2 f'' \cdot f'''\right) f\right]}{2\left(f' - 2\alpha f'' \cdot f''' \cdot f\right)^3} \tag{7}$$

El valor óptimo de α es $0,01$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$ en la raíz c.

El denominador del esquema $f' - 2\alpha f'' \cdot f''' \cdot f \neq 0$ en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación. Si la segunda derivada es igual a cero entonces el esquema se reduce al método de Newton.

Algoritmo 8 *Sea la función auxiliar $g(x) = e^{-\alpha \frac{f'}{f''}}$ entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{\left(f' - \alpha\left(1 - \dfrac{f' \cdot f'''}{f''^2}\right) f\right)} - \frac{f^2\left[\begin{array}{c} f'' - 2\alpha\left(1 - \dfrac{f' \cdot f'''}{f''^2}\right) f' + \\ \left(\begin{array}{c} \alpha\left(\dfrac{f'''}{f''} + \dfrac{f' \cdot f^{(4)}}{f''^2} - \dfrac{2f' \cdot f'''^2}{f''^3}\right) \\ +\alpha^2\left(1 - \dfrac{f' \cdot f'''}{f''^2}\right)^2 \end{array}\right) f \end{array}\right]}{2\left[f' - \alpha\left(1 - \dfrac{f' \cdot f'''}{f''^2}\right) f\right]^3} \tag{8}$$

El valor óptimo de α es $0,01$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$, $f''(c) \neq 0$ en la raíz c.

El denominador del esquema

$$f' - \alpha \left(1 - \frac{f' \cdot f'''}{f''^2}\right) f \neq 0$$

en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación.

Algoritmo 9 *Sea la función auxiliar* $g(x) = e^{-\alpha \frac{f}{f''}}$ *entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{f' - \alpha\left(\frac{f'}{f''} - \frac{f \cdot f'''}{f''^2}\right) f} - \frac{f^2 \left[\begin{array}{c} f'' - 2\alpha f' \cdot \left(\frac{f'}{f''} - \frac{f f'''}{f''^2}\right) + \\ \left(\begin{array}{c} -\alpha \left(\begin{array}{c} 1 - \frac{f \cdot f'''}{f''^2} - \frac{f' \cdot f''' + f \cdot f^{(4)}}{f''^2} \\ + \frac{2f \cdot f'''^2}{f''^3} \end{array} \right) \\ +\alpha^2 \left(\frac{f'}{f''} - \frac{f \cdot f'''}{f''^2}\right)^2 \end{array} \right) f \end{array} \right]}{2\left[f' - \alpha\left(\frac{f'}{f''} - \frac{f \cdot f'''}{f''^2}\right) f\right]^3} \tag{9}$$

El valor óptimo de α es $0,01$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$ y $f''(c) \neq 0$ en la raíz c.

El denominador del esquema

$$f' - \alpha \left(\frac{f'}{f''} - \frac{f \cdot f'''}{f''^2}\right) f \neq 0$$

en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación.

Algoritmo 10 *Sea la función auxiliar* $g(x) = e^{-\alpha \frac{f^3}{f'}}$ *entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{\left(f' - \alpha\left(\frac{3f^2}{f'} - \frac{f^3 f''}{f'^2}\right) f\right)} - \frac{f^2 \left[\begin{array}{c} f'' - 2\alpha \left(\frac{3f^2}{f'} - \frac{f^3 f''}{f'^2}\right) f' + \\ \left(\begin{array}{c} \alpha \left(\begin{array}{c} \frac{6f}{f'} - \frac{3f^3 \cdot f''}{f'^2} \\ - \frac{3f^2 \cdot f''}{f'^2} \\ - \frac{f^3 \cdot f'''}{f'^2} - \frac{2f^3 \cdot f''^2}{f'^3} \end{array} \right) \\ +\alpha^2 \left(\frac{3f^2}{f'} - \frac{f^3 f''}{f'^2}\right)^2 \end{array} \right) f \end{array} \right]}{2\left[f' - \alpha\left(\frac{3f^2}{f'} - \frac{f^3 f''}{f'^2}\right) f\right]^3} \tag{10}$$

Donde α es un número real positivo menor o igual a $0,001$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$ en la raíz c.

El denominador del esquema

$$f' - \alpha \left(\frac{3f^2}{f'} - \frac{f^3 f''}{f'^2} \right) f \neq 0$$

en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación.

Algoritmo 11 *Sea la función auxiliar* $g(x) = e^{-\alpha \frac{f \cdot f'}{f''}}$ *entonces el esquema iterativo toma la forma*

$$x_{n+1} = x_n - \frac{f}{f' - \alpha\left(f + \frac{f'^2}{f''} - \frac{f \cdot f' \cdot f'''}{f''^2}\right) f} - \frac{f^2 \left[\begin{array}{c} f'' - 2\alpha f' \left(f + \frac{f'^2}{f''} - \frac{f \cdot f' \cdot f'''}{f''^2}\right) \\ - \left(\begin{array}{c} \alpha f \left(\begin{array}{c} 3f' + \frac{f''^2}{f''} \\ - \frac{f'^2 \cdot f''' + f'^2 \cdot f'''}{f''^2} \\ - \frac{f \cdot f' \cdot f^{(4)}}{f''^2} \\ - \frac{2 \cdot f \cdot f' \cdot f'''^2}{f''^3} \end{array} \right) \\ + \alpha^2 \left(f + \frac{f'^2}{f''} - \frac{f \cdot f' \cdot f'''}{f''^2}\right)^2 f \end{array} \right) \end{array} \right]}{2\left[f' - \alpha\left(f + \frac{f'^2}{f''} - \frac{f \cdot f' \cdot f'''}{f''^2}\right) f\right]^3} \tag{11}$$

El valor óptimo de α es $0,01$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$ y $f''(c) \neq 0$ en la raíz c.

El denominador del esquema

$$f' - \alpha \left(f + \frac{f'^2}{f''} - \frac{f \cdot f' \cdot f'''}{f''^2} \right) f \neq 0$$

en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación.

Algoritmo 12 *Sea la función auxiliar* $g(x) = e^{-\alpha f' \cdot f''}$ *entonces el esquema toma la forma*

$$x_{n+1} = x_n - \frac{f}{\left(f' - \alpha\left(f''^2 + f' \cdot f'''\right) f\right)} - \frac{f^2 \left[\begin{array}{c} f'' - 2\alpha\left(f''^2 + f' \cdot f'''\right) f' + \\ \left(\begin{array}{c} -\alpha\left(3f'' \cdot f''' + f' \cdot f^{(4)}\right) \\ +\alpha^2 \left(f''^2 + f' \cdot f'''\right)^2 \end{array} \right) f \end{array} \right]}{2\left[f' - \alpha\left(f''^2 + f' \cdot f'''\right) f\right]^3} \tag{12}$$

Los valores óptimos de α son los números reales positivos menores que $0,01$. Para este algoritmo la función debe ser monótona, derivable y con $f'(c) \neq 0$. Si $f''(c) = f'''(c) = f^{(4)}(c) = 0$ en la raíz c entonces el método se reduce al método de Newton.

El denominador del esquema

$$f' - \alpha \left(f''^2 + f' \cdot f'''\right) f \neq 0$$

en este caso el esquema iterativo no se puede aplicar por la presencia de una indeterminación.

8.2. Variantes obtenidas mediante Polinomios de Adomian

El objetivo central de esta sección es considerar diferentes variantes de la función $N(h)$ y presentar el diseño de nuevos métodos iterativos, todos ellos basados únicamente en la técnica de descomposición de Adomian.

Dado que el método de Newton solo depende de una variable

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

Convirtiendo el método de Newton en un modelo bipaso

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

Que se puede expresar como

$$x_{n+1} = y_n - \Sigma A$$

Donde A_0 sería el primer polinomio de Adomian

$$A_0 = N(h_0)$$

$$A_0 = \frac{f(y_n)}{f'(y_n)}$$

y, además

$$h_0 = \frac{f(y_n)}{f'(y_n)}$$

Si se deriva A_1 para obtener el segundo término de los polinomios de Adomian.

$$A_1 = \frac{d}{d\lambda} N(h_0 + h_1\lambda)|_{\lambda=0} = h_1 N'\left(h_0 + h_1(0)\right) = h_1 N'(h_0)$$

$$A_1 = h_1 N'(h_0)$$

$$h_1 = \frac{f^2(y_n) \cdot f''(y_n)}{2\left[f'(y_n)\right]^3}$$

$$N'(h_0) = \frac{d}{dh_0} \frac{h_0^2}{2} \frac{f''(y_n)}{f'(y_n)} = \frac{2h_0}{2} \frac{f''(y_n)}{f'(y_n)} = h_0 \frac{f''(y_n)}{f'(y_n)} = \left(\frac{f(y_n)}{f'(y_n)}\right)\left(\frac{f''(y_n)}{f'(y_n)}\right) = \frac{f(y_n) \cdot f''(y_n)}{\left[f'(y_n)\right]^2}$$

Por lo tanto,

$$A_2 = h_1 N'(h_0) = \left(\frac{\left[f(y_n)\right]^2 f''(y_n)}{2\left[f'(y_n)\right]^3}\right)\left(\frac{f(y_n) f''(y_n)}{\left[f'(y_n)\right]^2}\right) = \frac{f(y_n)^3 \left[f''(y_n)\right]^2}{2\left[f'(y_n)\right]^5} = h_2$$

Nuevamente, para obtener el tercer coeficiente

$$A_3 = \frac{1}{2}\frac{d}{d\lambda}(h_1 + 2h_2\lambda) N'(h_0 + h_1\lambda)|_{\lambda=0}$$

$$A_3 = h_2 N'(h_0 + h_1\lambda)|_{\lambda=0} + \frac{1}{2} h_1 (h_1 + 2h_2\lambda) N'(h_0 + h_1\lambda)|_{\lambda=0}$$

$$A_3 = h_2 N'(h_0 + h_1\lambda)|_{\lambda=0} + \frac{1}{2} h_1 (h_1 + 2h_2\lambda) N''(h_0 + h_1\lambda)|_{\lambda=0}$$

$$A_3 = h_2 N'(h_0) + \frac{1}{2} h_1 h_1 N''(h_0)$$

$$A_3 = h_2 N'(h_0) + \frac{1}{2} h_1^2 N''(h_0)$$

Considerando que

$$h_0 = \frac{f(y_n)}{f'(y_n)};\ h_1 = \frac{\left[f(y_n)\right]^2 \cdot f''(y_n)}{2\left[f'(y_n)\right]^3};\ h_2 = \frac{\left[f(y_n)\right]^3 \cdot \left[f''(y_n)\right]^2}{2\left[f'(y_n)\right]^5}$$

Y, además

$$N'(h_0) = \frac{f(y_n) \cdot f''(y_n)}{\left[f'(y_n)\right]^2}$$

$$N''(h_0) = \frac{d}{dh_0}\left(h_0 \frac{f''(y_n)}{f'(y_n)}\right) = \frac{f''(y_n)}{f'(y_n)}$$

Por lo tanto, el tercer término de Adomian se obtiene mediante el siguiente razonamiento.

$$
\begin{aligned}
A_3 &= h_2 N'(h_0) + \frac{1}{2} h_1^2 N''(h_0) \\
A_3 &= \left(\frac{[f(y_n)]^2 \cdot [f''(y_n)]^2}{2\,[f'(y_n)]^5} \right) \left(\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2} \right) + \frac{1}{2} \left(\frac{[f(y_n)]^2 \cdot f''(y_n)}{2\,[f'(y_n)]^3} \right)^2 \left(\frac{f''(y_n)}{f'(y_n)} \right) \\
A_3 &= \frac{[f(y_n)]^4 \cdot [f''(y_n)]^3}{2\,[f'(y_n)]^7} + \frac{1}{2} \left(\frac{[f(y_n)]^4 \cdot [f''(y_n)]^2}{4\,[f'(y_n)]^6} \right) \left(\frac{f''(y_n)}{f'(y_n)} \right) \\
A_3 &= \frac{[f(y_n)]^4 \cdot [f''(y_n)]^3}{2\,[f'(y_n)]^7} + \frac{[f(y_n)]^4 \cdot [f''(y_n)]^3}{4\,[f'(y_n)]^7} = \frac{5}{8} \frac{[f(y_n)]^4 \cdot [f''(y_n)]^3}{4\,[f'(y_n)]^7} \\
A_3 &= h_3 = \frac{5}{8} \frac{[f(y_n)]^4 \cdot [f''(y_n)]^3}{[f'(y_n)]^7}
\end{aligned}
$$

Para el cuarto término, se sabe que

$$
\begin{aligned}
A_2 &= h_1 N'(h_0) \\
A_3 &= h_2 N'(h_0) + \frac{1}{2} h_1^2 N''(h_0)
\end{aligned}
$$

luego

$$A_4 = \frac{1}{3} \frac{d}{d\lambda} ((h_2 + 3h_3\lambda) N(h_0 + h_1\lambda) + \frac{1}{2} (h_1 + 2h_2\lambda)^2 N''(h_0 + h_1\lambda))|_{\lambda=0}$$

$$A_4 = \frac{1}{6} \left[6h_3 N'(h_0) + 6h_1 h_2 N''(h_0) + h_1^3 N'''(h_0) \right]$$

$$
\begin{aligned}
A_4 &= \frac{1}{3} \left[3h_3 N'(h_0 + h_1\lambda) + h_1(h_2 + 3h_3\lambda) N''(h_0 + h_1\lambda) \right] |_{\lambda=0} + \\
& \frac{1}{3} \left(\frac{1}{2} \right) \left[2(2h_2)(h_1 + 2h_2\lambda) N''(h_0 + h_1\lambda) + h_1 (h_1 + 2h_2\lambda)^2 N'''(h_0 + h_1\lambda) \right] |_{\lambda=0} \\
A_4 &= \frac{1}{3} \left[3h_3 N'(h_0) + h_1(h_2) N''(h_0) \right] + \left(\frac{1}{3} \right) \left(\frac{1}{2} \right) \left[(4h_2)(h_1) N''(h_0) + h_1(h_1^2) N''(h_0) \right] \\
A_4 &= \frac{1}{3} \left[3h_3 N'(h_0) + h_1 h_2 N''(h_0) \right] + \frac{1}{6} \left[2h_1 h_2 N''(h_0) + h_1^3 N'''(h_0) \right]
\end{aligned}
$$

$$A_4 = \frac{1}{6}\left[6h_3N'(h_0) + 2h_1h_2N''(h_0)\right] + \frac{1}{6}\left[4h_1h_2N''(h_0) + h_1^3N'''(h_0)\right]$$

$$A_4 = \frac{1}{6}\left[6h_3N'(h_0) + 6h_1h_2N''(h_0) + h_1^3N'''(h_0)\right]$$

Como

$$N'(h_0) = \frac{d}{dh_0}\left(\frac{h_0^2}{2}\frac{f''(y_n)}{f'(y_n)}\right) = \frac{2h_0}{2}\frac{f''(y_n)}{f'(y_n)} = h_0\frac{f''(y_n)}{f'(y_n)} = \frac{f(y_n)}{f'(y_n)}\frac{f''(y_n)}{f'(y_n)} = \frac{f(y_n)f''(y_n)}{\left[f'(y_n)\right]^2}$$

Entonces

$$N''(h_0) = \frac{d}{dh_0}\left(h_0\frac{f''(y_n)}{f'(y_n)}\right) = \frac{f''(y_n)}{f'(y_n)}$$

Por lo tanto,

$$N'(h_0) = \frac{f(y_n)\cdot f''(y_n)}{\left[f'(y_n)\right]^2}$$

$$N''(h_0) = \frac{d}{dh_0}\left(h_0\frac{f''(y_n)}{f'(y_n)}\right) = \frac{f''(y_n)}{f'(y_n)}$$

$$N'''(h_0) = \frac{d}{dh_0}\left(\frac{d}{dh_0}\left(h_0\frac{f''(y_n)}{f'(y_n)}\right)\right) = \frac{d}{dh_0}\left(\frac{f''(y_n)}{f'(y_n)}\right) = 0$$

Si se tiene presente los valores de h obtenidos anteriormente

$$h_1 = \frac{\left[f(y_n)\right]^2\cdot f''(y_n)}{2\left[f'(y_n)\right]^3}$$

$$h_2 = \frac{\left[f(y_n)\right]^3\cdot \left[f''(y_n)\right]^2}{2\left[f'(y_n)\right]^5}$$

$$h_3 = \frac{5}{8}\frac{\left[f(y_n)\right]^4\cdot \left[f''(y_n)\right]^3}{\left[f'(y_n)\right]^7}$$

Al sustituir resulta

$$A_4 = \frac{1}{6}\left[6h_3N'(h_0)+6h_1h_2N''(h_0)+h_1^3N'''(h_0)\right]$$

$$A_4 = \frac{1}{6}\left[\begin{array}{c} 6\left(\frac{5}{8}\right)\left(\frac{[f(y_n)]^4\cdot[f''(y_n)]^3}{[f'(y_n)]^7}\right)\left(\frac{f(y_n)\cdot f''(y_n)}{[f'(y_n)]^2}\right)+6\left(\frac{[f(y_n)]^2\cdot f''(y_n)}{2[f'(y_n)]^3}\right)\left(\frac{[f(y_n)]^3\cdot[f''(y_n)]^2}{2[f'(y_n)]^5}\right)\left(\frac{f''(y_n)}{f'(y_n)}\right) \\ +\left(\frac{[f(y_n)]^2\cdot f''(y_n)}{2[f'(y_n)]^3}\right)^3(0) \end{array}\right]$$

$$A_4 = \frac{1}{6}\left[\frac{30}{8}\left(\frac{[f(y_n)]^5\cdot[f''(y_n)]^4}{[f'(y_n)]^9}\right)+\frac{6}{4}\left(\frac{[f(y_n)]^5\cdot[f''(y_n)]^4}{[f'(y_n)]^9}\right)+0\right]$$

$$A_4 = \frac{1}{6}\left(\frac{42}{8}\frac{f^5\cdot f''^4}{f'^9}\right)=\frac{42}{48}\frac{[f(y_n)]^5\cdot[f''(y_n)]^4}{[f'(y_n)]^9}=\frac{7}{8}\frac{[f(y_n)]^5\cdot[f''(y_n)]^4}{[f'(y_n)]^9}$$

$$A_4 = h_4=\frac{7}{8}\frac{[f(y_n)]^5\cdot[f''(y_n)]^4}{[f'(y_n)]^9}$$

Para encontrar A_5 se tiene que

$$A_4=\frac{1}{6}\left[6h_3N'(h_0)+6h_1h_2N''(h_0)+h_1^3N'''(h_0)\right]$$

Por lo tanto

$$A_5 = \frac{1}{4}\frac{d}{d\lambda}A_4|_{\lambda=0}$$

$$A_5 = \frac{1}{24}\frac{d}{d\lambda}\left[\begin{array}{c} 6(h_3+4h_4\lambda)N'(h_0+h_1\lambda)+6(h_1+2h_2\lambda)(h_2+3h_3\lambda)N''(h_0+h_1\lambda)+ \\ (h_1+h_2\lambda)^3N''(h_0+h_1\lambda) \end{array}\right]|_{\lambda=0}$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 6\left[4h_4N'(h_0+h_1\lambda)+h_1(h_3+4h_4\lambda)N''(h_0+h_1\lambda)\right]+ \\ 6\left[\begin{array}{c} 2h_2(h_2+3h_3\lambda)N''(h_0+h_1\lambda)+3h_3(h_1+2h_2\lambda)N''(h_0+h_1\lambda)+ \\ h_1(h_1+2h_2\lambda)(h_2+3h_3\lambda)N'''(h_0+h_1\lambda) \end{array}\right] \\ +\left[3(2h_2)(h_1+2h_2\lambda)^2N'''(h_0+h_1\lambda)+h_1(h_1+2h_2\lambda)^3N^{(4)}(h_0+h_1\lambda)\right] \end{array}\right]|_{\lambda=0}$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 6\left[4h_4N'(h_0)+h_1(h_3)N'(h_0)\right]+6\left[\begin{array}{c} 2h_2(h_2)N''(h_0)+3h_3(h_1)N''(h_0)+ \\ h_1(h_1)(h_2)N'''(h_0) \end{array}\right]+ \\ \left[6h_2(h_1)^2N'''(h_0)+h_1(h_1)^3N^{(4)}(h_0)\right] \end{array}\right]$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 6\left[4h_4N'(h_0)+4h_1(h_3)N''(h_0)\right]+6\left[2h_2(h_2)N''(h_0)+2h_1(h_1)(h_2)N'''(h_0)\right]+ \\ \left[h_1(h_1)^3N^{(4)}(h_0)\right] \end{array}\right]$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 24h_4N'(h_0)+24h_1(h_3)N''(h_0)+12h_2^2N''(h_0)+48h_1(h_1)(h_2)N'''(h_0) \\ +h_1(h_1)^3N^{(4)}(h_0) \end{array}\right]$$

$$A_5 = \frac{1}{24}\left[24h_4N'(h_0)+(24h_1h_3+12h_2^2)N''(h_0)+12h_1^2h_2N'''(h_0)+h_1^4N^{(4)}(h_0)\right]$$

Hasta el momento se tiene que

$$N'(h_0) = \frac{f(y_n)\cdot f''(y_n)}{\left[f'(y_n)\right]^2}$$

$$N''(h_0) = \frac{d}{dh_0}\left(h_0\frac{f''(y_n)}{f'(y_n)}\right) = \frac{f''(y_n)}{f'(y_n)}$$

$$N'''(h_0) = \frac{d}{dh_0}\left(\frac{d}{dh_0}\left(h_0\frac{f''(y_n)}{f'(y_n)}\right)\right) = \frac{d}{dh_0}\left(\frac{f''(y_n)}{f'(y_n)}\right) = 0$$

$$N^{(4)}(h_0) = 0$$

y los valores de h *son*

$$h_1 = \frac{\left[f(y_n)\right]^2\cdot f''(y_n)}{2\left[f'(y_n)\right]^3}$$

$$h_2 = \frac{\left[f(y_n)\right]^3\cdot \left[f''(y_n)\right]^2}{2\left[f'(y_n)\right]^5}$$

$$h_3 = \frac{5}{8}\frac{\left[f(y_n)\right]^4\cdot \left[f''(y_n)\right]^3}{\left[f'(y_n)\right]^7}$$

$$h_4 = \frac{7}{8}\frac{\left[f(y_n)\right]^5\cdot \left[f''(y_n)\right]^4}{\left[f'(y_n)\right]^9}$$

y

$$A_5 = \frac{1}{24}\left[24h_4N'(h_0) + (24h_1h_3 + 12h_2^2)N''(h_0) + 12h_1^2h_2N'''(h_0) + h_1^4N^{(4)}(h_0)\right]$$

Como

$$N'''(h_0) = N^{(4)}(h_0) = 0$$

Se simplifica A_5 como

$$A_5 = \frac{1}{24}\left[24h_4N'(h_0) + (24h_1h_3 + 12h_2^2)N''(h_0)\right]$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 24\left(\frac{7}{8}\frac{[f(y_n)]^5\cdot[f''(y_n)]^4}{[f'(y_n)]^9}\right)\left(\frac{f(y_n)\cdot f''(y_n)}{[f'(y_n)]^2}\right)+ \\ \left(24\left(\frac{[f(y_n)]^2\cdot f''(y_n)}{2[f'(y_n)]^3}\right)\left(\frac{5}{8}\frac{[f(y_n)]^4\cdot[f''(y_n)]^3}{[f'(y_n)]^7}\right)+12\left(\frac{[f(y_n)]^3\cdot[f''(y_n)]^2}{2[f'(y_n)]^5}\right)^2\right)\left(\frac{f''(y_n)}{f'(y_n)}\right) \end{array}\right]$$

$$A_5 = \frac{1}{24}\left[\begin{array}{c} 24\left(\frac{7}{8}\right)\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^5}{[f'(y_n)]^{11}}\right) \\ +\left(\frac{120}{16}\frac{f^6\cdot f''^4}{f'^{10}}\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^4}{[f'(y_n)]^{10}}\right)+12\left(\frac{f^6\cdot f''^4}{4f'^{10}}\right)\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^4}{4[f'(y_n)]^{10}}\right)\right)\left(\frac{f''(y_n)}{f'(y_n)}\right) \end{array}\right]$$

$$A_5 = \frac{1}{24}\left[\frac{168}{8}\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^5}{[f'(y_n)]^{11}}\right)+\left(\begin{array}{c}\frac{60}{8}\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^4}{[f'(y_n)]^{10}}\right) \\ +3\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^4}{[f'(y_n)]^{10}}\right)\end{array}\right)\left(\frac{f''(y_n)}{f'(y_n)}\right)\right]$$

$$A_5 = \frac{1}{24}\left[\frac{168}{8}\frac{[f(y_n)]^6\cdot[f''(y_n)]^5}{[f'(y_n)]^{11}}+\left(\frac{84}{8}\frac{[f(y_n)]^6\cdot[f''(y_n)]^4}{[f'(y_n)]^{10}}\right)\left(\frac{f''(y_n)}{f'(y_n)}\right)\right]$$

$$A_5 = \frac{1}{24}\left[\frac{168}{8}\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^5}{[f'(y_n)]^{11}}\right)+\frac{84}{8}\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^5}{[f'(y_n)]^{11}}\right)\right]$$

$$A_5 = \frac{1}{24}\left[\frac{252}{8}\left(\frac{[f(y_n)]^6\cdot[f''(y_n)]^5}{[f'(y_n)]^{11}}\right)\right]$$

$$A_5 = h_5 = \frac{252}{192}\frac{[f(y_n)]^6\cdot[f''(y_n)]^5}{[f'(y_n)]^{11}} = \frac{21}{16}\frac{[f(y_n)]^6\cdot[f''(y_n)]^5}{[f'(y_n)]^{11}}$$

Como un análisis final se presenta el resumen de la discusión anterior, en forma de los polinomios de Adomian

$$
\begin{aligned}
A_0 &= \frac{f(y_n)}{f'(y_n)} \\
A_1 &= \frac{(f(y_n))^2 f''(y_n)}{2(f'(y_n))^3} \\
A_2 &= \frac{(f(y_n))\,(f''(y_n))^2}{2(f'(y_n))^5} \\
A_3 &= \frac{5}{8}\frac{(f(y_n))^2\,(f''(y_n))^3}{(f'(y_n))^7} \\
A_4 &= \frac{7}{8}\frac{(f(y_n))^5 (f''(y_n))^4}{(f'(y_n))^9} \\
A_5 &= \frac{21}{16}\frac{(f(y_n))^6 (f''(y_n))^5}{(f'(y_n))^{11}}
\end{aligned}
$$

Mediante los polinomios de Adomian, en un proceso bipaso se obtienen los siguientes algoritmos.

Método de Newton, considerando el segundo término de los polinomios de Adomian.

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
x_{n+1} &= y_n - A_0 - A_1
\end{aligned}
$$

Al sustituir se obtiene el siguiente algoritmo bipaso:

Algoritmo 13 *Sea la función $f(x)$ monótona, derivable, y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...$, tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$
y_n = x_n - \frac{f(x_n)}{f'(x_n)}
$$

$$
x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 f''(y_n)}{2(f'(y_n))^3} \tag{13}
$$

Si se considera el tercer término de los polinomios de Adomian

$$
y_n = x_n - \frac{f(x_n)}{f'(x_n)}
$$

$$
x_{n+1} = y_n - A_0 - A_1 - A_2
$$

Generando el algoritmo descrito a continuación

Algoritmo 14 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...,$tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 f''(y_n)}{2(f'(y_n))^3} - \frac{(f(y_n))^3 (f''(y_n))^2}{2(f'(y_n))^5} \tag{14}$$

Si se considera el tercer término de los polinomios de Adomian

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = y_n - A_0 - A_1 - A_2$$

Por lo tanto, la estructura se convierte en un algoritmo nuevo

Algoritmo 15 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...,$tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 f''(y_n)}{2(f'(y_n))^3} - \frac{(f(y_n))^3 (f''(y_n))^2}{2(f'(y_n))^5} \tag{15}$$

Si se considera el quinto término de los polinomios de Adomian

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = y_n - A_0 - A_1 - A_2 - A_3 - A_4$$

Algoritmo 16 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...,$tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 f''(y_n)}{2(f'(y_n))^3} - \frac{(f(y_n))^3 (f''(y_n))^2}{2(f'(y_n))^5} - \frac{(f(y_n))^3 (f''(y_n))^2}{2(f'(y_n))^5} - \frac{5}{8}\frac{(f(y_n))^4 (f''(y_n))^3}{(f'(y_n))^7} \tag{16}$$

Si se considera el sexto término de los polinomios de Adomian

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = y_n - A_0 - A_1 - A_2 - A_3 - A_4 - A_5$$

Algoritmo 17 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...,$tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$\begin{aligned} x_{n+1} &= y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 f''(y_n)}{2(f'(y_n))^3} - \frac{(f(y_n))^3 (f''(y_n))^2}{2(f'(y_n))^5} - \frac{(f(y_n))^3 (f''(y_n))^2}{2(f'(y_n))^5} \\ &\quad - \frac{5}{8}\frac{(f(y_n))^4 (f''(y_n))^3}{(f'(y_n))^7} - \frac{7}{8}\frac{(f(y_n))^5 (f''(y_n))^4}{(f'(y_n))^9} - \frac{21}{16}\frac{(f(y_n))^6 (f''(y_n))^5}{(f'(y_n))^{11}} \end{aligned} \tag{17}$$

Ahora se propone que donde A_0 sería el primer polinomio de Adomian

$$A_0 = N(h_0)$$

$$A_0 = \frac{f(y_n)}{f'(y_n)}$$

y, además,

$$h_0 = \frac{f(y_n)}{f'(y_n)}$$

Si se deriva A_1 del polinomio de Adomian para obtener el segundo término, de la misma forma como se mostró anteriormente

$$A_2 = h_1 N'(h_0)$$

Se propone que $A_1 = h_1 = f(y_n)$.Dado que

$$N'(h_0) = \frac{d}{dh_0}\frac{h_0^2}{2}\frac{f''(y_n)}{f'(y_n)} = \frac{2h_0}{2}\frac{f''(y_n)}{f'(y_n)} = h_0\frac{f''(y_n)}{f'(y_n)} = \left(\frac{f(y_n)}{f'(y_n)}\right)\left(\frac{f''(y_n)}{f'(y_n)}\right) = \frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}$$

Por lo tanto

$$\begin{aligned} A_2 &= h_1 N'(h_0) \\ A_2 &= f(y_n)\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2} = \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} \end{aligned}$$

Dado que los algoritmos se forman en la forma

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

y para la variable x_n

$$x_n = y_n - A_0 - A_1 - A_2$$

es decir

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - f(y_n) - \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2}$$

Teniendo en cuenta que $f(y_n) \approx 0$ entonces $A_1 \approx 0$ por lo tanto, se puede descartar.

Algoritmo 18 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...$,tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_n = y_n - -\frac{f(y_n)}{f'(y_n)} - \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} \tag{18}$$

Continuando con el razonamiento anterior

$$N''(h_0) = \frac{d}{dh_0}\left(h_0 \frac{f''(y_n)}{f'(y_n)}\right) = \frac{f''(y_n)}{f'(y_n)}$$

Los siguientes datos muestran que

$$\begin{aligned} h_1 &= f(y_n) \\ h_2 &= f'(y_n) \\ N'(h_0) &= \frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2} \\ N''(h_0) &= \frac{f''(y_n)}{f'(y_n)} \end{aligned}$$

Además se ha demostrado que el tercer término de Adomian viene dado por

$$\begin{aligned} A_3 &= h_2 N'(h_0) + \frac{1}{2} h_1^2 N''(h_0) \\ A_3 &= f'(y_n)\left(\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}\right) + \frac{1}{2}(f(y_n))^2\left(\frac{f''(y_n)}{f'(y_n)}\right) \\ A_3 &= \frac{f(y_n) \cdot f''(y_n)}{f'(y_n)} + \frac{1}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)} \\ A_3 &= \frac{3}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)} \end{aligned}$$

Dado que los algoritmos se forman en la forma

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

y para la variable x_n

$$x_n = y_n - A_0 - A_1 - A_2 - A_3$$

Como $f(y_n) \approx 0$, entonces $A_1 \approx 0$, por lo tanto se puede simplificar,

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{3}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)}$$

Se presenta el siguiente algoritmo

Algoritmo 19 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...$,tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{3}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)} \tag{19}$$

Continuando con el razonamiento anterior

$$N'''(h_0) = \frac{d}{dh_0}\left(\frac{f''(y_n)}{f'(y_n)}\right) = 0$$

Las ecuaciones anteriores permiten decir que $h_1 = \alpha f(y_n)$ y,

$$\begin{aligned} h_1 &= \alpha f(y_n) \\ h_2 &= \alpha f'(y_n) \\ h_3 &= \alpha f''(y_n) \end{aligned}$$

$$N'(h_0) = \frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}$$

$$N''(h_0) = \frac{f''(y_n)}{f'(y_n)}$$

$$N'''(h_0) = 0$$

Queda demostrado que el cuarto término de Adomian viene dado por

$$\begin{aligned} A_4 &= \frac{1}{6}\left[6h_3N'(h_0) + 6h_1h_2N''(h_0) + h_1^3N''(h_0)\right] \\ A_4 &= \frac{1}{6}\left[6\alpha f''(y_n)\left(\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}\right) + 6\left(\alpha f(y_n)\right)\left(\alpha f'(y_n)\right)\left(\frac{f''(y_n)}{f'(y_n)}\right) + \left(\alpha f(y_n)\right)^3(0)\right] \\ A_4 &= \frac{1}{6}\left[6\alpha f''(y_n)\left(\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}\right) + 6\alpha^2 f(y_n)\left(f''(y_n)\right)\right] \end{aligned}$$

Como $f(y_n) \approx 0$ entonces $A_1 \approx 0$ por esta razón se puede simplificar el segundo término

$$A_4 = \frac{1}{6}\left[6\frac{\alpha f(y_n) \cdot (f''(y_n))^2}{[f'(y_n)]^2}\right] = \frac{\alpha f(y_n) \cdot (f''(y_n))^2}{[f'(y_n)]^2}$$

Dado que los algoritmos se forman en la forma

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

y para la variable x_n

$$x_n = y_n - A_0 - A_1 - A_2 - A_3 - A_4$$

Como $f(y_n) \approx 0$ y $A_1 \approx 0$ esto implica que se puede simplificar, es decir

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{\alpha\,[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{3}{2}\frac{\alpha f(y_n)^2 f''(y_n)}{f'(y_n)} - \frac{\alpha f(y_n) \cdot (f''(y_n))^2}{[f'(y_n)]^2}$$

Si se supone los siguientes valores

$$\begin{aligned} h_0 &= \frac{f(y_n)}{f'(y_n)} \\ h_1 &= f(y_n) \end{aligned}$$

Luego se realiza el siguiente cambio en el valor de h_2 igualándolo con el valor de A_2

$$\begin{aligned} h_2 &= \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} \\ N'(h_0) &= \frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2} \\ N''(h_0) &= \frac{f''(y_n)}{f'(y_n)} \end{aligned}$$

Además, se ha demostrado que el tercer término de Adomian viene dado por

$$\begin{aligned} A_3 &= h_2 N'(h_0) + \frac{1}{2} h_1^2 N''(h_0) \\ A_3 &= \left(\frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2}\right)\left(\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}\right) + \frac{1}{2}\left(f(y_n)\right)^2 \left(\frac{f''(y_n)}{f'(y_n)}\right) \\ A_3 &= \frac{(f(y_n))^3 \cdot (f''(y_n))^2}{(f'(y_n))^4} + \frac{1}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)} \end{aligned}$$

Dado que los algoritmos se forman en la forma

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

y para la variable x_n

$$x_n = y_n - A_0 - A_1 - A_2 - A_3$$

Como $f(y_n) \approx 0$ entonces $A_1 \approx 0$ se puede simplificar

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{(f(y_n))^3 \cdot (f''(y_n))^2}{(f'(y_n))^4} - \frac{1}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)}$$

Se presenta el siguiente algoritmo

Algoritmo 20 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...$,tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{(f(y_n))^3 \cdot (f''(y_n))^2}{(f'(y_n))^4} - \frac{1}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)} \tag{20}$$

Razonando de igual forma

$$\begin{aligned} h_0 &= \frac{f(y_n)}{f'(y_n)} \\ h_1 &= f(y_n) \end{aligned}$$

$$\begin{aligned} h_2 &= \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} \\ h_3 &= \frac{(f(y_n))^3 \cdot (f''(y_n))^2}{(f'(y_n))^4} + \frac{1}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)} \\ N'(h_0) &= \frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2} \\ N''(h_0) &= \frac{f''(y_n)}{f'(y_n)} \end{aligned}$$

El tercer término de Adomian es

$$\begin{aligned} A_4 &= \frac{1}{6}\left[6h_3N'(h_0) + 6h_1h_2N''(h_0) + h_1^3N'''(h_0)\right] \\ A_4 &= \frac{1}{6}\left[6\left(\frac{(f(y_n))^3 \cdot (f''(y_n))^2}{(f'(y_n))^4} + \frac{1}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)}\right)\left(\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}\right)\right] \\ &\quad + \frac{1}{6}\left[+6f(y_n)\left(\frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2}\right)\left(\frac{f''(y_n)}{f'(y_n)}\right) + f(y_n)^3(0)\right] \\ A_4 &= \frac{1}{6}\left[6\left(\frac{(f(y_n))^4 \cdot (f''(y_n))^3}{(f'(y_n))^4} + \frac{1}{2}\frac{f(y_n)^3\,(f''(y_n))^2}{[f'(y_n)]^3}\right) + \left(\frac{6\,[f(y_n)]^3 \cdot (f''(y_n))^2}{[f'(y_n)]^3}\right)\right] \\ A_4 &= \frac{(f(y_n))^4 \cdot (f''(y_n))^3}{(f'(y_n))^4} + \frac{f(y_n)^3\,(f''(y_n))^2}{2\,[f'(y_n)]^3} + \frac{[f(y_n)]^3 \cdot (f''(y_n))^2}{[f'(y_n)]^3} \end{aligned}$$

Los algoritmos presentan la forma

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

y para la variable x_n

$$x_n = y_n - A_0 - A_1 - A_2 - A_3$$

Como $f(y_n) \approx 0$ entonces, $A_1 \approx 0$ y se puede considerar cero este término

$$\begin{aligned} x_n &= y_n - \frac{f(y_n)}{f'(y_n)} - \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{(f(y_n))^3 \cdot (f''(y_n))^2}{(f'(y_n))^4} - \frac{1}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)} \\ &\quad - \frac{(f(y_n))^4 \cdot (f''(y_n))^3}{(f'(y_n))^4} - \frac{f(y_n)^3\,(f''(y_n))^2}{2\,[f'(y_n)]^3} - \frac{[f(y_n)]^3 \cdot (f''(y_n))^2}{[f'(y_n)]^3} \end{aligned}$$

Se presenta el siguiente algoritmo

Algoritmo 21 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...$,tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$\begin{aligned} x_n &= y_n - \frac{f(y_n)}{f'(y_n)} - \frac{[f(y_n)]^2 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{(f(y_n))^3 \cdot (f''(y_n))^2}{(f'(y_n))^4} - \frac{1}{2}\frac{f(y_n)^2 f''(y_n)}{f'(y_n)} \\ &\quad - \frac{(f(y_n))^4 \cdot (f''(y_n))^3}{(f'(y_n))^4} - \frac{3f(y_n)^3\,(f''(y_n))^2}{2\,[f'(y_n)]^3} \end{aligned} \tag{21}$$

Se considera la siguiente condición $h_1 = f(y_n)f'(y_n)$, entonces

$$\begin{aligned} h_0 &= \frac{f(y_n)}{f'(y_n)} \\ A_0 &= \frac{f(y_n)}{f'(y_n)} \end{aligned}$$

Para el primer término se tiene que

$$\begin{aligned} h_1 &= f(y_n)f'(yn) \\ A_1 &= f(y_n)f'(yn) \end{aligned}$$

Y el segundo término de Adomian sería

$$A_2 = h_1 N'(h_0)$$

Mientras tanto los valores anteriores, se siguen conservando

$$N'(h_0) = \frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}$$

Entonces,

$$\begin{aligned} A_2 &= f(y_n)f'(y_n)\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2} \\ A_2 &= \frac{(f(y_n))^2 \cdot f''(y_n)}{f'(y_n)} \end{aligned}$$

De igual forma,

$$h_2 = \frac{(f(y_n))^2 \cdot f''(y_n)}{f'(y_n)}$$

La sección del algoritmo relativa a x_n es

$$x_n = y_n - A_0 - A_1 - A_2$$

Como $A_1 = f(y_n)f'(y_n)$ y, además $f(y_n) \approx 0$ entonces, $A_1 \approx 0$ y se simplifica de la siguiente forma

$$x_n = y_n - A_0 - A_2$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 \cdot f''(y_n)}{f'(y_n)}$$

Por lo tanto, el nuevo algoritmo es

Algoritmo 22 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...,$tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 \cdot f''(y_n)}{f'(y_n)} \tag{22}$$

Se considera la siguiente condición $h_1 = f(y_n)y_n$, entonces

$$\begin{aligned} h_0 &= \frac{f(y_n)}{f'(y_n)} \\ A_0 &= \frac{f(y_n)}{f'(y_n)} \end{aligned}$$

Para el primer término se tiene que

$$\begin{aligned} h_1 &= f(y_n)y_n \\ A_1 &= f(y_n)y_n \end{aligned}$$

Y el segundo término de Adomian sería

$$A_2 = h_1 N'(h_0)$$

Mientras tanto los valores anteriores, se siguen conservando

$$N'(h_0) = \frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}$$

Entonces

$$\begin{aligned} A_2 &= f(y_n)y_n \frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2} \\ A_2 &= \frac{y_n (f(y_n))^2 \cdot f''(y_n)}{[f'(y_n)]^2} \end{aligned}$$

La sección del algoritmo relativa a x_n es

$$x_n = y_n - A_0 - A_1 - A_2$$

Si $A_1 = f(y_n)y_n$y, $f(y_n) \approx 0$ se concluye que $A_1 \approx 0$ y se simplifica de la siguiente forma

$$x_n = y_n - A_0 - A_2$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{y_n\left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2}$$

Algoritmo 23 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...,$tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{y_n\left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2} \tag{23}$$

Continuando con el razonamiento anterior

$$N''(h_0) = \frac{d}{dh_0}\left(h_0\frac{f''(y_n)}{f'(y_n)}\right) = \frac{f''(y_n)}{f'(y_n)}$$

En resumen,

$$\begin{aligned} h_1 &= f(y_n)y_n \\ h_2 &= \frac{y_n\left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2} \\ N'(h_0) &= \frac{f(y_n) \cdot f''(y_n)}{\left[f'(y_n)\right]^2} \\ N''(h_0) &= \frac{f''(y_n)}{f'(y_n)} \end{aligned}$$

El tercer término de Adomian se calcula mediante el siguiente razonamiento

$$\begin{aligned} A_3 &= h_2N'(h_0) + \frac{1}{2}h_1^2N''(h_0) \\ A_3 &= \left(\frac{y_n\left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2}\right)\left(\frac{f(y_n) \cdot f''(y_n)}{\left[f'(y_n)\right]^2}\right) + \frac{1}{2}\left(y_nf(y_n)\right)^2\left(\frac{f''(y_n)}{f'(y_n)}\right) \\ A_3 &= \frac{y_n\left(f(y_n)\right)^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^4} + \frac{1}{2}\frac{\left(y_n\right)^2 f(y_n)^2 f''(y_n)}{f'(y_n)} \end{aligned}$$

Luego

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

y para la variable x_n

$$x_n = y_n - A_0 - A_1 - A_2 - A_3$$

Se observa $y_n f(y_n) \approx 0$ entonces $A_1 \approx 0$ se puede simplificar, es decir

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{y_n\left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2} - \frac{y_n\left(f(y_n)\right)^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^4} - \frac{1}{2}\frac{\left(y_n\right)^2 f(y_n)^2 f''(y_n)}{f'(y_n)}$$

Algoritmo 24 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...,$tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{y_n\left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2} - \frac{y_n\left(f(y_n)\right)^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^4} - \frac{1}{2}\frac{\left(y_n\right)^2 f(y_n)^2 f''(y_n)}{f'(y_n)} \tag{24}$$

Teniendo en cuenta que

$$\begin{aligned}
h_1 &= f(y_n) y_n \\
h_2 &= \frac{y_n\left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2} \\
h_3 &= \frac{y_n\left(f(y_n)\right)^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^4} + \frac{1}{2}\frac{\left(y_n\right)^2 f(y_n)^2 f''(y_n)}{f'(y_n)} \\
N'(h_0) &= \frac{f(y_n) \cdot f''(y_n)}{\left[f'(y_n)\right]^2} \\
N''(h_0) &= \frac{f''(y_n)}{f'(y_n)} \\
N'''(h_0) &= 0
\end{aligned}$$

Además, se ha demostrado que el tercer término de Adomian viene dado por

$$A_4 = \frac{1}{6}\left[6h_3 N'(h_0) + 6h_1 h_2 N''(h_0) + h_1^3 N'''(h_0)\right]$$

Al sustituir

$$\begin{aligned}
A_4 &= \frac{1}{6}\left[6h_3 N'(h_0) + 6h_1 h_2 N''(h_0) + h_1^3 N'''(h_0)\right] \\
A_4 &= \frac{1}{6}\left[\begin{array}{c} 6\left(\frac{y_n(f(y_n))^3 \cdot (f''(y_n))^2}{[f'(y_n)]^4} + \frac{1}{2}\frac{(y_n)^2 f(y_n)^2 f''(y_n)}{f'(y_n)}\right)\left(\frac{f(y_n) \cdot f''(y_n)}{[f'(y_n)]^2}\right) \\ +6y_n f(y_n)\left(\frac{y_n(f(y_n))^2 \cdot f''(y_n)}{[f'(y_n)]^2}\right)\left(\frac{f''(y_n)}{f'(y_n)}\right) + (y_n f(y_n))^3(0) \end{array}\right] \\
A_4 &= \frac{1}{6}\left[6\left(\frac{y_n\left(f(y_n)\right)^4 \cdot \left(f''(y_n)\right)^3}{\left(f'(y_n)\right)^6} + \frac{1}{2}\frac{\left(y_n\right)^2 f(y_n)^3 \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^3}\right) + \left(\frac{6\left(y_n\right)^2\left[f(y_n)\right]^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^3}\right)\right] \\
A_4 &= \frac{y_n\left(f(y_n)\right)^4 \cdot \left(f''(y_n)\right)^3}{\left(f'(y_n)\right)^6} + \frac{\left(y_n\right)^2 f(y_n)^3 \left(f''(y_n)\right)^2}{2\left[f'(y_n)\right]^3} + \frac{\left(y_n\right)^2\left[f(y_n)\right]^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^3} \\
A_4 &= \frac{y_n\left(f(y_n)\right)^4 \cdot \left(f''(y_n)\right)^3}{\left(f'(y_n)\right)^6} + \frac{3}{2}\frac{\left(y_n\right)^2 f(y_n)^3 \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^3}
\end{aligned}$$

La forma base del valor de y_n

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

y para x_n

$$x_n = y_n - A_0 - A_1 - A_2 - A_3$$

Dado $y_n f(y_n) \approx 0$ entonces $A_1 \approx 0$, se puede simplificar

$$\begin{aligned} x_n &= y_n - \frac{f(y_n)}{f'(y_n)} - \frac{y_n \left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2} - \frac{y_n \left(f(y_n)\right)^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^4} - \frac{1}{2}\frac{\left(y_n\right)^2 f(y_n)^2 f''(y_n)}{f'(y_n)} \\ &\quad - \frac{y_n \left(f(y_n)\right)^4 \cdot \left(f''(y_n)\right)^3}{\left(f'(y_n)\right)^6} - \frac{3}{2}\frac{\left(y_n\right)^2 \left[f(y_n)\right]^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^3} \end{aligned}$$

Se presenta el siguiente algoritmo

Algoritmo 25 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...$,tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$\begin{aligned} x_n &= y_n - \frac{f(y_n)}{f'(y_n)} - \frac{y_n \left(f(y_n)\right)^2 \cdot f''(y_n)}{\left[f'(y_n)\right]^2} - \frac{y_n \left(f(y_n)\right)^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^4} - \frac{1}{2}\frac{\left(y_n\right)^2 f(y_n)^2 f''(y_n)}{f'(y_n)} \\ &\quad - \frac{y_n \left(f(y_n)\right)^4 \cdot \left(f''(y_n)\right)^3}{\left(f'(y_n)\right)^6} - \frac{3}{2}\frac{\left(y_n\right)^2 \left[f(y_n)\right]^3 \cdot \left(f''(y_n)\right)^2}{\left[f'(y_n)\right]^3} \end{aligned} \tag{25}$$

Manteniendo el valor de $h_0 = \frac{f(y_n)}{f'(y_n)}$, pero proponiendo que $h_1 = \left[f(y_n)\right]^2$,

$$\begin{aligned} h_0 &= \frac{f(y_n)}{f'(y_n)} \\ A_0 &= \frac{f(y_n)}{f'(y_n)} \end{aligned}$$

El primer término es

$$\begin{aligned} h_1 &= \left[f(y_n)\right]^2 \\ A_1 &= \left[f(y_n)\right]^2 \end{aligned}$$

El segundo término

$$A_2 = h_1 N'(h_0)$$

Los valores de $N'(h_0)$ únicamente dependen del valor de h_0, así que se usan los valores obtenidos anteriormente

$$N'(h_0) = \frac{f(y_n) \cdot f''(y_n)}{\left[f'(y_n)\right]^2}$$

Y el valor de A_2

$$\begin{aligned} A_2 &= [f(y_n)]^2 \frac{f(y_n)\cdot f''(y_n)}{[f'(y_n)]^2} \\ A_2 &= \frac{(f(y_n))^3\cdot f''(y_n)}{[f'(y_n)]^2} \end{aligned}$$

La sección del algoritmo relativa a x_n es

$$x_n = y_n - A_0 - A_1 - A_2$$

Si $A_1 = [f(y_n)]^2$pero a medida que $f(y_n) \to 0$, entonces $A_1 \to 0$ y resulta

$$x_n = y_n - A_0 - A_2$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^3\cdot f''(y_n)}{[f'(y_n)]^2}$$

Algoritmo 26 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...,$tomando un valor semilla cercano a la raíz x_0, la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^3\cdot f''(y_n)}{[f'(y_n)]^2} \tag{26}$$

A partir del análisis inicial

$$N''(h_0) = \frac{d}{dh_0}\left(h_0\frac{f''(y_n)}{f'(y_n)}\right) = \frac{f''(y_n)}{f'(y_n)}$$

$$\begin{aligned} h_1 &= (f(y_n))^2 \\ h_2 &= \frac{(f(y_n))^3\cdot f''(y_n)}{[f'(y_n)]^2} \\ N'(h_0) &= \frac{f(y_n)\cdot f''(y_n)}{[f'(y_n)]^2} \\ N''(h_0) &= \frac{f''(y_n)}{f'(y_n)} \end{aligned}$$

El tercer término de Adomian es

$$\begin{aligned} A_3 &= h_2 N'(h_0) + \frac{1}{2}h_1^2 N''(h_0) \\ A_3 &= \left(\frac{(f(y_n))^3\cdot f''(y_n)}{[f'(y_n)]^2}\right)\left(\frac{f(y_n)\cdot f''(y_n)}{[f'(y_n)]^2}\right) + \frac{1}{2}\left((f(y_n))^2\right)^2\left(\frac{f''(y_n)}{f'(y_n)}\right) \\ A_3 &= \frac{(f(y_n))^4\cdot (f''(y_n))^2}{[f'(y_n)]^3} + \frac{1}{2}\frac{(f(y_n))^4\, f''(y_n)}{f'(y_n)} \end{aligned}$$

Luego

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

y para la variable x_n

$$x_n = y_n - A_0 - A_1 - A_2 - A_3$$

Como $A_1 = [f(y_n)]^2$, pero a medida que $f(y_n) \to 0$, entonces $A_1 \to 0$ se concluye

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^3 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{(f(y_n))^4 \cdot (f''(y_n))^2}{[f'(y_n)]^3} - \frac{1}{2}\frac{(f(y_n))^4 f''(y_n)}{f'(y_n)}$$

Algoritmo 27 *Sea la función $f(x)$ monótona, derivable y con $f'(x) \neq 0$, con $n = 0, 1, 2, 3, ...$, tomando un valor semilla cercano a la solución x_0 la solución de la ecuación $f(x) = 0$ puede encontrarse con*

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^3 \cdot f''(y_n)}{[f'(y_n)]^2} - \frac{(f(y_n))^4 \cdot (f''(y_n))^2}{[f'(y_n)]^3} - \frac{1}{2}\frac{(f(y_n))^4 f''(y_n)}{f'(y_n)} \tag{27}$$

9. CAPÍTULO V: ANÁLISIS Y DISCUSIÓN DE RESULTADOS

En este acápite se presentan algunos ejemplos donde se ilustra la eficiencia de los nuevos métodos desarrollados. La comparación se realiza entre el método de Newton y los algoritmos encontrados. Se tomará como base fundamental el número de iteraciones para encontrar dicha raíz. El conjunto de funciones bases utilizadas, son aquellas funciones que fueron consideradas en diferentes artículos de investigación, las cuales poseen características de ser funciones continuas y diferenciables.

Se han programado todos los algoritmos desarrollados en el lenguaje de alto nivel Python, bajo la filosofía de Programación Orientado a Objeto (POO) y en todos los criterios de parada de los códigos fuente, se utiliza un nivel de tolerancia de 10^{-15}. Este nivel de tolerancia ha sido usado anteriormente por Noor (2007, p. 130).

Tabla 1: Pasos a seguir para la codificación de los algoritmos.
Fuente: Elaboración Propia

Inicio	
Entrada	Introducir función de la cual se requiere buscar una raíz
	Introducir función primera Derivada y función segunda derivada
	Introducir el valor semilla
Paso 1	Contador=0 #(Cuenta la cantidad de iteraciones)
Paso 2	Mientras: Sea verdadero
Paso 3	Introducir algoritmo
Paso 4	Si: error$\leq 10^{-15}$ #La ecuación de error es la ecuación del error relativo.
Paso 5	Imprimir xn
Paso 6	Fin Si
Paso 7	Añadir una unidad al contador #(Añade la iteración nueva)
Paso 8	Fin

El pseudocódigo generado en PSeint se usa para reflejar los pasos seguidos, previos a la programación, en Python el método de Newton y la ecuación $e^{-x} - x^3 = 0$.

```
Funcion res <- f(x)
    res<-exp(-x)-x^3
FinFuncion
```

```
Funcion res <-f1(x)
	res<-(-1)*exp(-x)-3*x^2
FinFuncion

Proceso newton
	conta=0
	x=1.0
	Repetir
		x1=x-f(x)/f1(x)
		e=abs((x1-x)/x1)
		conta=conta+1
		x=x1
	Hasta Que e<=10^(-15)
Escribir "la solución es: ",x1,.en ", conta,"iteraciones"
FinProceso
```

El pseudocódigo para el algoritmo 1 en pseint, es:

```
Funcion res <- f(x)
	res<-exp(-x)-x^3
FinFuncion

Funcion res <-f1(x)
	res<-(-1)*exp(-x)-3*x^2
FinFuncion

Funcion res<-f2(x)
	res<-exp(-x)-6*x
FinFuncion

Proceso algoritmo_1
	conta=0
	x=1.0
	a=0.01
	Repetir
		x1=x-f(x)/(f1(x)-a*f(x))-(f(x))^2*(f2(x)-2*a*f1(x)*f(x)+a^2*f(x))/(2*(f1(x)-a*f(x))^3)
		e=abs((x1-x)/x1)
```

```
        conta=conta+1
        x=x1
    Hasta Que e<=10^(-15)
Escribir "la solución es: " ,x1," en ",conta," iteraciones"
FinProceso
```

Un ejemplo reducido para la programación en Python mediante clase para algoritmos seleccionados (método de Newton, Algoritmo Cisneros Méndez 1 y 2, y algoritmo 17) es el siguiente:

```
#Se importan el módulo a utilizar, especificamente para invocar la función exponencial
import numpy as np

#Se nombra la clase
class Iterativos:
```

#Se define cada una de la función principal $f(x)$ de $f(x) = 0$

```
def f(self,x):
    return (np.exp(-x)-x**3)
```

#Se define la primer derivada $f'(x)$

```
def f1(self,x):
    return (-np.exp(-x)-3*x**2)
```

#Se define la segunda derivada $f''(x)$

```
def f2(self,x):
    return (np.exp(-x)-6*x)
```

#Se define la función método de Newton.

```
def Método_de_Newton(self, x0):
    conta=0 #Esto contará el número de iteraciones
    error=10e-15 #Esto marca el error dado
    while True: #Se debe iterar el algoritmo mientras el error sea mayor que el error dado
        xn=x0-self.f(x0)/self.f1(x0) #Este es el algoritmo a aplicar
        if abs(xn-x0)/xn<=error: #El error relativo debe ser menor o igual al error
            print("La solución es obtenida por Newton en ", conta, "iteraciones {:.15f}".format(xn))
            break #Si se cumple la condición el algoritmo debe detenerse
```

```
        conta+=1 #Si la condición no se cumple se añade uno al contador de iteraciones
        x0=xn #Si la condición no se cumple se reemplaza el valor anterior por el nuevo
    return #Terminar la ejecución
```

#Se define el algoritmo Cisneros Méndez 1

```
def Cisneros_Méndez1(self, x0):
    conta=0
    while True:
        xnn=x0-self.f(x0)/self.f1(x0)-(1/2)*(self.f(x0))**2*(self.f2(x0))/(self.f1(x0)**3)
        -(1/2)*(self.f(x0))**3*(self.f2(x0))**2/(self.f1(x0)**5)
        -(5/8)*(self.f(x0))**4*(self.f2(x0))**3/(self.f1(x0)**7)
        if abs(xnn-x0)/xnn<=10e-15:
            print("La solución obtenida en ", conta, "iteraciones es  {:.15f}".format(xnn))
            break
        conta+=1
        x0=xnn
    return
```

#Se define el algoritmo Cisneros Méndez 2

```
def Cisneros_Méndez2(self, x0):
    conta=0
    while True:
        xnn=x0-self.f(x0)/self.f1(x0)-(1/2)*(self.f(x0))**2*(self.f2(x0))/(self.f1(x0)**3)
        -(1/2)*(self.f(x0))**3*(self.f2(x0))**2/(self.f1(x0)**5)
        -(5/8)*(self.f(x0))**4*(self.f2(x0))**3/(self.f1(x0)**7)
        -(7/8)*(self.f(x0))**5*(self.f2(x0))**4/(self.f1(x0)**9)
         if abs(xnn-x0)/xnn<=10e-15:
            print("La solución obtenida en ", conta, "iteraciones es {:.15f}".format(xnn))
            break
        conta+=1
        x0=xnn
     return
```

#Algoritmo 17, Este es un algoritmo bipaso

```
def Algoritmo17(self,x0):
    conta=0
    error=10e-15
    while True:
        yn=x0-self.f(x0)/self.f1(x0)
        #Se añade la función para encontrar y_n
        xn=yn-self.f(yn)/self.f1(yn)-self.f(yn)**2*self.f2(yn)/self.f1(yn)**2
        if abs(xn-yn)/xn<=error: #El error relativo para modelos bipaso.
            print("La solución por el algoritmo 17 en ", conta, "iteraciones es {:.15f}".format(xn))
            break
        conta+=1
        x0=xn
    return

iterativos=Iterativos()
x0=1.0 #Inserta el valor semilla
print("\nMÉTODO DE NEWTON\n")
print(iterativos.Método_de_Newton(x0))
print("VARIANTES OBTENIDAS POR POLINOMIOS DE ADOMIAN")
print(iterativos.Cisneros_Méndez1(x0))
print(iterativos.Cisneros_Méndez2(x0))
print(iterativos.Algoritmo17(x0))
```

El proceso de selección de los algoritmos fue sencillo, primero se enumeraron todos los algoritmos de manera secuencial y se programaron mediante clases, luego fueron seleccionados únicamente aquellos que verifican la condición de ejecutarse en un número menor de iteraciones que el método de Newton y aquellos que logran el mismo número de iteración, con un valor semilla específico. Esto se realiza con el objetivo de comparar los algoritmos óptimos y los que son equivalentes (en el número de iteraciones) al método de Newton.

Se aclara que la significación de equivalencia de método en esta tesis, significa que la raíz de la ecuación no lineal converge en un número determinado de iteraciones, aunque su expresión algebraica difiera en la estructura de su fórmula.

Como ejemplos numéricos se examinarán las ecuaciones usadas por Cisneros (2017), para probar sus algoritmos.

Ejemplo 1 *Se considera la ecuación no lineal* $e^{-x} - x^3 = 0$

Los principales hallazgos respecto a esta ecuación no lineal son:

El método de Newton alcanza la raíz de dicha ecuación en 5 iteraciones, siendo la raíz real con 15 cifras decimales $0,772882959149210$. Se toma como valor semilla a $x = 1,0$

El algoritmo 13, 14, 15, 16, 17, 18, 21, 23, 24, 26 y 27 genera la solución en 2 iteraciones.

El algoritmo Cisneros-Méndez 1, Cisneros-Méndez 2, Cisneros-Méndez 3, 1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 19, 20 y 25 generan la solución en tres iteraciones, superando al método de Newton.

El algoritmo 5 y 22 genera la solución en 4 iteraciones, superando al método de Newton en una iteración.

Ejemplo 2 *Se considera la ecuación no lineal* $x^3 + 4x^2 - 10 = 0$

Esta ecuación fue usada por Cisneros (2017) y además por Bumbariu (2012, p.277) como ejemplo numérico.

Los principales hallazgos respecto a esta ecuación no lineal son:

El método de Newton alcanza la raíz de dicha ecuación en 5 iteraciones, siendo la raíz real con 15 cifras decimales $1,365230013414097$.

Se toma como valor semilla a $x = 1,2$. Los algoritmos 13, 14, 15, 16, 17, 18, 23, 16 y 27 generan la solución en dos iteraciones.

El algoritmo Cisneros-Méndez 1, Cisneros-Méndez 2, Cisneros-Méndez 3, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 19, 21, 22, 24 y 25 generan la solución en 3 iteraciones.

Los algoritmos 12 y 20 generan la solución en 4 iteraciones.

Ejemplo 3 *Se considera la ecuación no lineal* $e^{-x} + 2\ln x = 0$

Los principales hallazgos respecto a esta ecuación no lineal son:

El método de Newton alcanza la raíz de dicha ecuación en 6 iteraciones, siendo la raíz real con 15 cifras decimales $0,798518085322260$. Se toma como valor semilla a $x = 1,5$.

Los algoritmos 13 y 14 generan la solución en 2 iteraciones.

Los algoritmos Cisneros-Méndez 1, Cisneros-Méndez 3, 1, 2, 3, 4, 6, 9, 10, 11, 12, 15, 16, 17, 18, 19, 23, 24, 25, 26 y 27 generan la solución en 3 iteraciones, superando al método de Newton en dos iteraciones.

Los algoritmos Cisneros-Méndez 2, 5, 7, 8 y 22 generan la solución en 4 iteraciones, superando al método de Newton en dos iteraciones.

Los algoritmos 20 y 21 presentan en la segunda iteración un valor negativo, que impide seguir corriendo el algoritmo por la característica del logaritmo natural que solo acepta números reales positivos como argumento, para salvar esta dificultad es necesario elegir un valor semilla menor tal como 1,3, desde este valor, el método de Newton genera una solución en 5 iteraciones, y los algoritmos 20 y 21 desde este valor semilla lo genera en 3 iteraciones, superando al método de Newton en dos iteraciones.

Si se evalúan los demás algoritmos en el valor semilla 1,3, el Método de Newton alcanza la solución en la quinta iteración, se obtiene que los algoritmos 13, 14 y 19, generan la solución en la segunda iteración; los algoritmos Cisneros-Méndez 1 y 3, los algoritmos 1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26 y 27 requieren tres iteraciones. Los algoritmos Cisneros-Méndez 2, el algoritmo 5 y el 8, alcanzan la solución a la cuarta iteración, superando al método de Newton en una iteración.

Es importante resaltar que las ecuaciones ($f(x) = 0$) donde se utilizan los algoritmos debe de poseer cierto número de derivadas (funciones diferenciables de un orden específico, especialmente el orden 2), determinadas por la naturaleza del algoritmo. Esto incluye que las funciones $f(x)$ deben ser continuas en el entorno entre, el valor semilla y la raíz exacta, y las funciones deben ser monótonas. En el caso de que el denominador fuera cero, o al menos una iteración en el proceso, obligue al denominador de las fracciones presentes en el algoritmos a anularse, entonces, el algoritmo no puede aplicarse.

Los algoritmos se pueden clasificar según el orden de derivación necesario:

1. Algoritmos de clase 2: Estos algoritmos requieren del cálculo de la primera y segunda derivada es decir en funciones de al menos C^2 en $\mathbb{R}$.

Los algoritmos de este tipo son Cisneros-Méndez 1, Cisneros-Méndez 2, Cisneros-Méndez 3, Algoritmo 1, 2, 3, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27.

2. Algoritmos de clase 3: los algoritmos de este tipo requieren del cálculo de la primera, segunda y tercera derivada, en funciones C^3 en $\mathbb{R}$.

   Los algoritmos de este tipo son el 4, 7, 10, 12.

3. Algoritmos de clase 4: los cuales necesitan del cálculo de la primera segunda, tercera y cuarta derivada, funciones en C^4 en $\mathbb{R}$.

   Los algoritmos de este tipo son 5, 6, 8, 9, 11.

   En una tabla se puede resumir:

Tabla 2: Clasificación de los Algoritmos.
Fuente: Elaboración Propia

Funciones en las que puede aplicarse	Funciones de Clase 2	Funciones de Clase 3	Funciones de Clase 4
Algoritmos	Cisneros Méndez 1, 2 y 3 Algoritmo 1, 2, 3, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,26, 27	4, 7, 10, 12	5, 6, 8, 9, 11
Total	21	4	5

10. CONCLUSIONES

Las conclusiones referentes a la presente investigación se pueden sistematizar en las siguientes aseveraciones:

- Generación de nuevos esquemas iterativos que representan treinta nuevas variantes del método de Newton por medio de la descomposición de los polinomios de Adomian y la combinación de la Técnica Iterativa Variacional.

- Existencia de nuevos algoritmos, que constituyen variantes del método de Newton que superan al método clásico en el número de iteraciones necesarias para la obtención de una raíz simple.

- Los nuevos algoritmos obtenidos son aplicaciones directas de haber considerado el método de Abbasbandy, con un desarrollo de la serie de Taylor de orden 2 y 3 respectivamente.

- Los algoritmos Cisneros-Méndez 1, Cisneros-Méndez 2, Cisneros-Méndez 3; los algoritmos 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 23, 24, 25, 26 y 27 superan al método de Newton en al menos una iteración, si el valor semilla es cercano a la solución real de la solución.

- En el conjunto de funciones base, se muestra la existencia de algoritmos que son equivalentes al método de Newton en la cantidad de iteraciones para lograr la raíz real.

- Todos los algoritmos obtenidos por estas variantes del método de Newton son convergentes, para las ecuaciones$e^{-x} - x^3 = 0$, $x^3 + 4x^2 - 10 = 0$, $e^{-x} + 2\ln x = 0$, oscilando todos ellos entre 3 y 5 iteraciones como máximo para valores semillas cercanos a la solución real.

- Obtención de diversos polinomios de Adomian para la generación de nuevos métodos iterativos.

11. RECOMENDACIONES

- Promover en los cursos de postgrados a nivel de maestría y doctorado en Matemática Aplicada el estudio de los métodos numéricos.

- Incluir capacitaciones referentes al estudio de lenguaje de programación de alto nivel con la filosofía del paradigma de programación orientada a objeto (POO).

- Actualización del acervo bibliográfico del programa de maestría y doctorado de Matemática Aplicada.

- Promover en las carreras de grado universitario, específicamente en Física Matemática, la publicación de resultados investigativos.

- Combinar métodos numéricos que puedan dar origen a nuevas líneas de investigación referente a hallazgos de soluciones de ecuaciones y sistemas no lineales.

- Buscar nuevas familias de funciones auxiliares para la creación de nuevos algoritmos, basado en la descomposición de Adomian y las Técnicas Iterativas Variacional.

12. BIBLIOGRAFÍA

1. Abbasbandy, S. (2003). Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method.Applied Mathematics and Computation 145 (2003) 487-893.

2. Amat. S., Busquiera, S. & Gutiérrez, J. (2003). Geometric constructions of iterative functions to solve nonlinear equations. Journal of Computational and Applied Mathematics, 197-205.

3. Arévalo, D., Bernal, M. & Posada, J. (2017). Matemáticas para Ingeniería, Métodos numéricos con Python. Bogotá: Editorial Politécnico Grancolombiano.

4. Azagra, D. (2018). Lo esencial sobre funciones de clase C, polinomios de Taylor, extremos y extremos condicionados. Recuperado el 09 de Marzo de 2023, de Lo esencial sobre funciones de clase C, polinomios de Taylor, extremos y extremos condicionados: http://www.mat.ucm.es/~dazagrar/do

5. Bahgat, M. & Hafiz, M. (2012). Solving nonsmooth equations using family. Journal of the Egyptian Mathematical Society, 38-43.

6. Bahgat, M. & Hafiz, M. (2014). Three-step iterative method with eighteenth order convergence for solving nonlinear equations. International Journal of Pure and Applied Mathematics Volume 93 No. 1, 85-94.

7. Bartle, R. & Sherbert, D. (2004). Introducción al análisis matemático de una variable (Segunda ed.). Limusa Wiley.

8. Basto, M., Semiao, V., Calheiros, F. (2005). A new iterative method to compute nonlinear equations. Applied Mathematics and Computation 173 (2006) 468–483.

9. Bi, W., Ren, H. & Wu, Q. (2009). Three-step iterative methods with eighth-order convergence for solving nonlinear equations. Journal of Computational and Applied Mathematics, 105-112.

10. Bumbariu, O. (2012). An Acceleration technique for slowly convergent fixed point iterative methods. Miskolc Mathematical Notes Vol. 13 (2012), No. 2, 271–281.

11. Burden, R., Faires, D. & Burden, A. (2017). Análisis Numérico. (Décima ed.). Cengage Learning.

12. Ceballos, J. (2007). C, C++ Curso de Programación (Tercera ed.). Madrid: Ra-Ma.

13. Chun, C. (2005). Iterative Methods Improving Newton's Method by the Decomposition Method. Computers and Mathematics with Applications 50 (2005), 1559-1568.

14. Chun, C. (2006). A family of composite fourth-order iterative methods for solving nonlinear equations. Applied Mathematics and Computation xxx (2006), 1-5.

15. Cisneros, I. (2017). Algoritmos basados en los Polinomios de Adomian e Interación Variacional para la resolución de ecuaciones no lineales. Recuperado el 24 de junio de 2021, de Algoritmos basados en los Polinomios de Adomian e Interación Variacional para la resolución de ecuaciones no lineales: https://repositorio.unan.edu.ni/11014/

16. Cordero, A., Torregrosa, J. & Vassileva, M. (2011). Three-step iterative methods with optimal eighth-order convergence. Journal of Computational and Applied Mathematics, 3189–3194.

17. Curia, M., Manterola, M., Medrano, M., Páez, N. y Wachenchauzer, R. (2011). Algoritmos de programación con Phyton. Recuperado el 23 de Marzo de 2021, de Algoritmos de programación con Phyton: https://uniwebsidad.com/libros/algoritmos-python.

18. Díaz-Barrero, J. & Grau. M. (2003). An improvement of the Euler-Chevyshev iterative method. Journal of mathematical analysis and applications, 1-7.

19. Herrera, A., & Cisneros, I. (2022). Algoritmización para la resolución de ecuaciones no lineales mediante la técnica de iteración variacional. Revista Científica De FAREM-Estelí, 11(41), 60–76. Recuperado el 02 de marzo del 2023 de Algoritmización para la resolución de ecuaciones no lineales mediante la técnica de iteración variacional: https://doi.org/10.5377/farem.v11i41.13886.

20. Ganji, D., Nourollahi, M. & Rostamian, M. (2006). A Comparison of Variational Iteration Method with Adomian's Decomposition Method in Some Highly Nonlinear Equations. International Journal of Science & Technology Volume 106, No. 1 2016-1, 1-10.

21. González, R. (2015). Python para todos. Recuperado el 08 de Marzo de 2023, de Python para todos: https://persoal.citius.usc.es/eva.cernadas/informaticaparacientificos/material/libros/Python%20

22. Guzdial, M. y Ericson, B. (2013). Introducción a la computación y programación con Python (Tercera ed.). México: Pearson Education.

23. Johanes, L., Fernández, M., y Rodriguez, L. (2003). Fundamentos de Programación. Libro de Problemas. (Segunda ed.). Madrid: McGrawHill.

24. Karthikeyan, K. (2016). Alternate Iterative Numerical Algorithms For Minimization Of Unconstrained Nonlinear Functions. International Journal of Science & Technology Volume 1, No 2, 179-188.

25. Karthikeyan, K. (2016). Alternate Iterative Numerical Algorithms For Minimization Of Unconstrained Nonlinear Functions. International Journal of Science & Technology Volume 2, No 2, 1-10.

26. Kou J., Li, Y. & Wang X. (2006). Third-order modification of Newton's method. Journal of Computational and Applied Mathematics, 1-5.

27. Koua, J., Lib Y. & Wang, X. (2006). Some variants of Ostrowski's method with seventh-order convergence. Journal of Computational and Applied Mathematics, 153–159.

28. Kumar, M., Singh, A. & Srivastava, A. (2013). Various Newton-type iterative methods for solving. Journal of the Egyptian Mathematical Society, 334-339.

29. Lizárraga, C. & Shingareva, I. (2012). Método De Descomposición De Adomian: Soluciones Analíticas Aproximadas De Ecuaciones Diferenciales Parciales No Lineales. Memorias de la XXII Semana de Investigación Nivel: Superior y Docencia en Matemáticas. Departamento de Matemáticas, Marzo, 2012, pp. 21–27.

30. Montalvo, F. (2003). Funciones de clase C. Recuperado el 09 de Marzo de 2023, de Funciones de clase C: https://matematicas.unex.es/~montalvo/Analisis_Varias_Variables/apuntes/cap07.pdf

31. Noor, M. (2007). New classes of iterative methods for nonlinear equations. Applied Mathematics and Computation 191 (2007), 128–131.

32. Noor, M., Noor, K. & Aftad, K. (2012). Some New Iterative Methods for Solving Nonlinear Equations. World Applied Sciences Journal 20 (6), 870-874.

33. Ozban, A. (2004). Some New Variants of Newton's Method. Applied Mathematics Letters 17 (2004), 677-682.

34. Petkovic, M. & Thukral, R. (2010). A family of three-point methods of optimal order for solving nonlinear equations. Journal of Computational and Applied Mathematics, 2278–2284.

35. Rossum, G. (2017). Tutorial de Python. Recuperado el 05 de Marzo de 2022, de Tutorial de Python: http://tutorial.python.org.ar/

36. Rudin, W. (1980). Principios de análisis matemático (Tercera ed.). México: McGraw Hill.

37. Samarski, A. (1988). Introducción a los Métodos Numéricos.(Primera ed.). Moscú: Editorial Mir.

38. Shah, F. (2012). Variational Iteration Technique and Numerical Methods for Solving Nonlinear Equations [Tesis doctoral]. Islamabab: COMSATS Institute of Information Technology.

39. Sharma, J. (2003). A Family of Methods for Solving Nonlinear Equations Using Quadratic Interpolation. Computers and Mathematics with Applications 48 (2004), 709-714.

40. Subrahmanyan, P. & Vijesh, V. (2007). A variant of Newton's method and its application. Journal of mathematical analysis and applications, 361-370.

41. Swoskoskwi & Cole . (2009). Algebra y trigonometría con geometría analitica. México, D.F: Centage Learning.

42. Thukral, R. (2016). New Modification of Newton Method with Third-Order Convergence for Solving Nonlinear Equations of Type f(x)=0. American Journal of Computational and Applied Mathematics 2016, 6(1), 14-18.

43. Trejos, O. (1999). La Esencia de la lógica de Programación. Pereira: Editorial Papiro.

44. Vadillo, J. (2019). Aprende Python desde cero a experto. Recuperado el 23 de Marzo de 2021, de Aprende Python desde cero a experto: http://leanpub.com/aprende-python

45. Verma, K. (2016). On a Variant of Newton's Method for Simple and Multiple Roots of Non Linear Equations. International Journal of Applied Science and Engineering , 101-114.

46. Weerakoon, S. & Fernando, T. (1998). A Variant of Newton's Method with Accelerated Third-Order Convergence. Applied Mathematics Letters 13 (2000) , 87-93.

47. Zill, D. (1996). Cálculo con Geometría Analítica. México: Prentice Hall.

13. ANEXO

Método de Newton

El método clásico para resolver ecuaciones no lineales lo constituye el método de Newton, el cual está definido por los siguientes algoritmos.

Algoritmo 1

Dado un valor inicial x_0 *se puede aproximar a la solución* x_{n+1} *mediante el esquema iterativo*

$$x_{n+1} = x_n - \frac{f\left(x_n\right)}{f'\left(x_n\right)} \quad , \quad f'\left(x_n\right) \neq 0 \ , \ n = 0, 1, 2, \cdots$$

este método tiene orden de convergencia cuadrática.

El siguiente algoritmo se utiliza para ecuaciones no lineales cuyas raíces poseen un orden de multiplicidad m.

Algoritmo 2

Dado un valor inicial x_0 *se puede aproximar a la solución* x_{n+1} *mediante el esquema iterativo*

$$x_{n+1} = x_n - m\frac{f\left(x_n\right)}{f'\left(x_n\right)} \quad , \quad f'\left(x_n\right) \neq 0 \ , \ n = 0, 1, 2, \cdots$$

donde m *es el orden de multiplicidad de la ecuación no lineal. Este método tiene orden de convergencia cuadrática.*

Cuando se desconoce la multiplicidad de los ceros, se puede utilizar el siguiente algoritmo.

Algoritmo 3

Dado un valor inicial x_0 *se puede aproximar a la solución* x_{n+1} *mediante el esquema iterativo*

$$x_{n+1} = x_n - \frac{f\left(x_n\right) f'\left(x_n\right)}{\left[f'\left(x_n\right)\right]^2 - f\left(x_n\right) f'\left(x_n\right)} \quad , \quad \left[f'\left(x_n\right)\right]^2 \neq f\left(x_n\right) f'\left(x_n\right) \ , \ n = 0, 1, 2, \cdots$$

donde m *es el orden de multiplicidad del cero de la ecuación no lineal.*

Polinomios de Adomian

Los polinomios de Adomian están dados por:

$$A_n = \frac{1}{n!}\frac{d^n}{d\lambda^n}\left[N\left(\sum_{i=0}^{\infty} \lambda^i h_i\right)\right]_{\lambda=0} \quad , \quad n = 0, 1, 2, 3, \cdots \tag{28}$$

Los primeros cuatros polinomios de Adomian son

$$
\begin{aligned}
A_0 &= N\left(\sum_{i=0}^{\infty} \lambda^i h_i\right)_{\lambda=0} = N\left(h_0\right) \\
A_1 &= \frac{d}{d\lambda}\left[N\left(\sum_{i=0}^{\infty} \lambda^i h_i\right)\right]_{\lambda=0} = h_1 N'\left(h_0\right) \\
A_2 &= \frac{1}{2}\frac{d^2}{d\lambda^2}\left[N\left(\sum_{i=0}^{\infty} \lambda^i h_i\right)\right]_{\lambda=0} = \frac{1}{2}h_1^2 N''\left(h_0\right) + h_2 N'\left(h_0\right) \\
A_3 &= \frac{1}{6}\frac{d^3}{d\lambda^3}\left[N\left(\sum_{i=0}^{\infty} \lambda^i h_i\right)\right]_{\lambda=0} = \frac{1}{6}\left[h_1^3 N'''\left(h_0\right) + 5h_1 h_2 N''\left(h_0\right) + h_3 N'\left(h_0\right)\right]
\end{aligned}
$$

Los cuales pueden ser expresados en función de las derivadas por

$$
\begin{aligned}
h_0 &= \frac{f\left(x_n\right)}{f'\left(x_n\right)} = c \\
h_1 &= N\left(h_0\right) = \frac{h_0^2}{2}\frac{f''\left(x_n\right)}{f'\left(x_n\right)} = \frac{f^2\left(x_n\right) f''\left(x_n\right)}{2\left[f'\left(x_n\right)\right]^3} = A_0 \\
h_2 &= h_1 N'\left(h_0\right) = h_1 h_0 \frac{f''\left(x_n\right)}{f'\left(x_n\right)} = \frac{f^3\left(x_n\right)\left[f''\left(x_n\right)\right]^2}{2\left[f'\left(x_n\right)\right]^5} = A_1 \\
h_3 &= \frac{1}{2}h_1^2 N''\left(h_0\right) + h_2 N\left(h_0\right) = \frac{3}{8}\frac{f^5\left(x_n\right)}{\left[\left(f'\left(x_n\right)\right]^8}\left[f''\left(x_n\right)\right]^3 = A_2 \\
h_4 &= \frac{1}{6}\left[h_1^3 N'''\left(h_0\right) + 5h_1 h_2 N''\left(h_0\right) + h_3 N'\left(h_0\right)\right] = \frac{1}{32}\frac{f^5\left(x_n\right)}{\left[f'\left(x_n\right)\right]^9}\left[f''\left(x_n\right)\right]^4 = A_3
\end{aligned}
$$

Como $h = c + N\left(h\right)$ se tiene

$$\sum_{n=0}^{\infty} h_n = c + \sum_{n=0}^{\infty} A_n$$

Por ser la serie h convergente, se tiene la siguiente importante relación

$$h_0 = c \quad y \quad h_{n+1} = A_n \;\; para \; n \geq 0$$

Esquemas Iterativos Complementarios

1. **Algoritmo del Método de Newton**

   Dado x_0 valor semilla

   Para $n = 1, 2, \cdots$, hacer $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)} \quad , \quad f'(x_n) \neq 0$

   Generar una sucesión de valores x_{n+1} y verificar la cota de error permisible

2. **Método Iterativo de Abbs**

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n) f''(x_n)}{2[f'(x_n)]^3} - \frac{f^3(x_n)[f''(x_n)]^2}{2[f'(x_n)]^5} \quad ,. \quad n \geq 0$$

3. **Método Iterativo de Chun**

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)} \\ \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)} - 2\dfrac{f(y_n)}{f'(x_n)} + \dfrac{f(y_n) f'(y_n)}{[f'(x_n)]^2} \quad , \quad n \geq 0 \end{cases}$$

4. **Método de Halley**

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

5. **Método de HouseHolder (Método 1)**

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

6. **Método de Wu**

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)} \\ \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f'(y_n)} \end{cases}$$

7. **Método Iterativo de Abbasbandy (1 forma)**

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)} \\ \\ x_{n+1} = y_n - \dfrac{f^2(x_n) f''(x_n)}{2[f'(x_n)]^3} - \dfrac{f^3(x_n)[f''(x_n)]^2}{2[f'(x_n)]^5} \end{cases}$$

8. **Método Iterativo de Abbasbandy (2 forma)**

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)} \\ \\ x_{n+1} = y_n - \dfrac{f^2(x_n)\, f''(x_n)}{2f'(x_n)} - \dfrac{f^3(x_n)\, f'''(x_n)}{6\left[f'(x_n)\right]^4} \end{cases}$$

9. **Método de Ostrowsky**

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)} \\ \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f'(x_n)} \left(\dfrac{f(y_n)}{f(x_n) - 2f(y_n)} \right) \end{cases}$$

10. **Método Iterativo de Chun (1 forma)**

$$\begin{cases} y_k = x_n - \dfrac{f(x_n)}{f'(x_n)} \\ \\ x_{nk+1} = y_k - 2\dfrac{f(y_n)}{f'(x_n)} + \dfrac{f(y_n)\, f'(y_n)}{\left[f'(x_n)\right]^2} \end{cases}$$

11. **Método Iterativo de Chun (2 forma)**

$$\begin{cases} y_k = x_n - \dfrac{f(x_n)}{f'(x_n)} \\ \\ x_{nk+1} = y_k - \dfrac{f(y_n)}{f'(x_n)} \end{cases}$$