Universidad Internacional de La Rioja (UNIR)

Máster universitario en Ingeniería Matemática y Computación

Desarrollo y análisis de un nuevo método iterativo para ecuaciones no lineales llamado Danby-Castillo

Trabajo Fin de Maestría en Matemáticas y Computación

Presentado por: Castillo Gámez José Luis

Director/a: Jorba Cusco Marc

Ciudad: Guayaquil
Fecha: 22 de julio de 2020

Prólogo y Agradecimientos

El presente trabajo con nombre "Desarrollo y análisis de un nuevo método iterativo para ecuaciones no lineales llamado Danby-Castillo" es una propuesta que entra en el ámbito de investigación pura.

Desde que cursé la maestría quería realizar un trabajo, el cual trate de mejorar algún desarrollo o crear un nuevo modelo matemático para resolver de manera más eficiente algún problema en particular. La idea para la creación de este nuevo método necesitó algún tiempo de indagación en libros y trabajos de investigación para no solo realizar un método nuevo, sino que también hacerlo mejor a métodos convencionales, lo cual no es tan fácil y es más complejo que presentar un proyecto enfocado a una aplicación real o desarrollo práctico. Me produce mucha satisfacción aportar con un grano de arena en este tan amplio campo de la ciencia como es el de las matemáticas en el desarrollo de un nuevo método, el que otras personas podrán estudiar.

Quiero agradecer a mi madre por darme toda la confianza, sin ella no hubiera sido posible esto, a mi padre por los consejos y a mis familiares por el apoyo. A mi tutor por estar siempre cuando lo necesitaba y a la UNIR por brindarme estos valiosos conocimientos.

Resumen

Entre los diversos problemas matemáticos tenemos las ecuaciones no lineales, las cuales no todas tienen un método analítico para poderlas resolver de manera directa, en estos casos se usan los métodos iterativos que dependiendo de ciertos criterios pueden tener resultados satisfactorios o excelentes. En la actualidad hay una gran cantidad de métodos iterativos, pero con el continuo avance de la ciencia siguen aumentando y en muchos casos mejorando a los anteriores. El propósito de este TFM es crear un nuevo proceso iterativo, el cual tenga un mayor orden de convergencia y eficiencia al compararse con el método del cual surgió la idea, en este caso del método iterativo de Danby, que es uno de los métodos de un punto con más alto orden de convergencia. Para facilitar cálculos se realizarán los algoritmos de los métodos con el software Matlab R2018a. Al final encontraremos las conclusiones, ventajas, desventajas, líneas de trabajo futuro y referencias bibliográficas. El resultado presentado nos mostrará lo eficiente que es el nuevo método en la tabla de comparación de resultados.

Palabras claves:

Esquema iterativo, orden de convergencia, métodos de un punto, algoritmo.

Abstract

Among the various math problems, we have non-linear equations, which not all have an analytical method to be able to solve them directly, in these cases the iterative methods are used, which depending on certain criteria can have satisfactory or excellent results. Currently there are a lot of iterative methods, but with the continuous advancement of science they continue to increase and improve on the previous ones. The purpose of this TFM is to create a new iterative process, which has a greater order of convergence and efficiency compared to the method from which the idea arose in this case, the Danby iterative method which is one of the highest convergence order of one-point methods. To facilitate calculations, the algorithms of the methods will be performed with the Matlab R2018a software. At the end we will find the conclusions, their advantages, disadvantages, future lines of work and bibliographic references. The result presented will show us how efficient the new method is in the results comparison table.

Keyword:

Iterative scheme, order of convergence, one-point methods, algorithm.

Índice de contenido

1. Introducción 1
1.1. Justificación.. 3
1.2. Planteamiento del trabajo 3
1.3. Estructura de la memoria... 4
2. Revisión y estado del arte 5
3. Objetivos concretos 8
3.1 Objetivo general 8
3.2 Objetivos específicos.. 8
4. Marco teórico.. 9
4.1 El inicio hacia el uso de métodos iterativos 9
4.2 Métodos iterativos de punto fijo 10
4.2.1 Convergencia.. 13
4.2.2 Índices de eficiencia 15
4.3 Biografía de Danby 19
4.3.1 Realización de su método... 20
4.3.2 Orden de convergencia local del método 21
5. Desarrollo de la contribución... 23
5.1. Realización del método propuesto 23
5.1.1. Convergencia local del método propuesto 25
5.2. Resolución a un problema concreto 27
5.2.1. Resolución por el método de Danby 28
5.2.2. Resolución por el método propuesto.. 39
5.3. Elaboración del algoritmo del método propuesto 45
6. Conclusiones 47
6.1 Ventajas.. 49
6.2 Desventajas.. 50
6.3 Líneas futuras 50
7. Bibliografía 52
8. Anexos 56
8.1 Programa de resolución para el método de Danby..................................... 56

8.2 Programa de resolución para el método propuesto 57

8.3 Comparación de resultados 59

Índice de tablas

Tabla 1. “Resultados de iteraciones y errores absoluto y porcentual para el método de Danby”..36
Tabla 2. “Resultado del ACOC por cada iteración para el método de Danby ”..................37
Tabla 3. “Resultados de iteraciones y errores absoluto y porcentual para el método propuesto” 43
Tabla 4. “Resultado del ACOC por cada iteración para el método propuesto” 43

Índice de figuras

Figura 1. “Gráfica a la solución de una ecuación no lineal con diferentes raíces”. 1
Figura 2. “Clasificación de los métodos iterativos” 2
Figura 3. “Modelo general del método iterativo punto fijo” 11
Figura 4. “Muestras de gráficos de diferentes tasas de convergencia” 17
Figura 5. “Representación gráfica de tipos de soluciones a sistemas de ecuaciones”.........19
Figura 6. “Representación del método de Newton” 24
Figura 7. “Gráfica del problema” 29
Figura 8. “Gráfica de la función del problema” 30
Figura 9. “Representación gráfica de la tasa de convergencia del método de Danby” 38
Figura 10. “Representación gráfica de la tasa de convergencia del método propuesto”.....44

1. Introducción

En matemáticas e ingeniería es común encontrarnos con ecuaciones no lineales de la forma $f(x) = 0$ (en donde a x se le conoce como el cero de la función), después de haber seguido un cierto procedimiento o también puede que en algún problema la ecuación este planteada directamente.

Dada una función f, la cual sea m veces diferenciable y cumpla f: $I \subset R \rightarrow R$, decimos que:

- x^* es una raíz de la función f con multiplicidad m si cumple que:

$$f(x^*) = f'(x^*) = f''(x^*) = \dots = f^{(m-1)}(x^*) = 0 \text{ y } f^{(m)}(x^*) \neq 0$$

en el caso de m = 1, se dice que la raíz es simple.

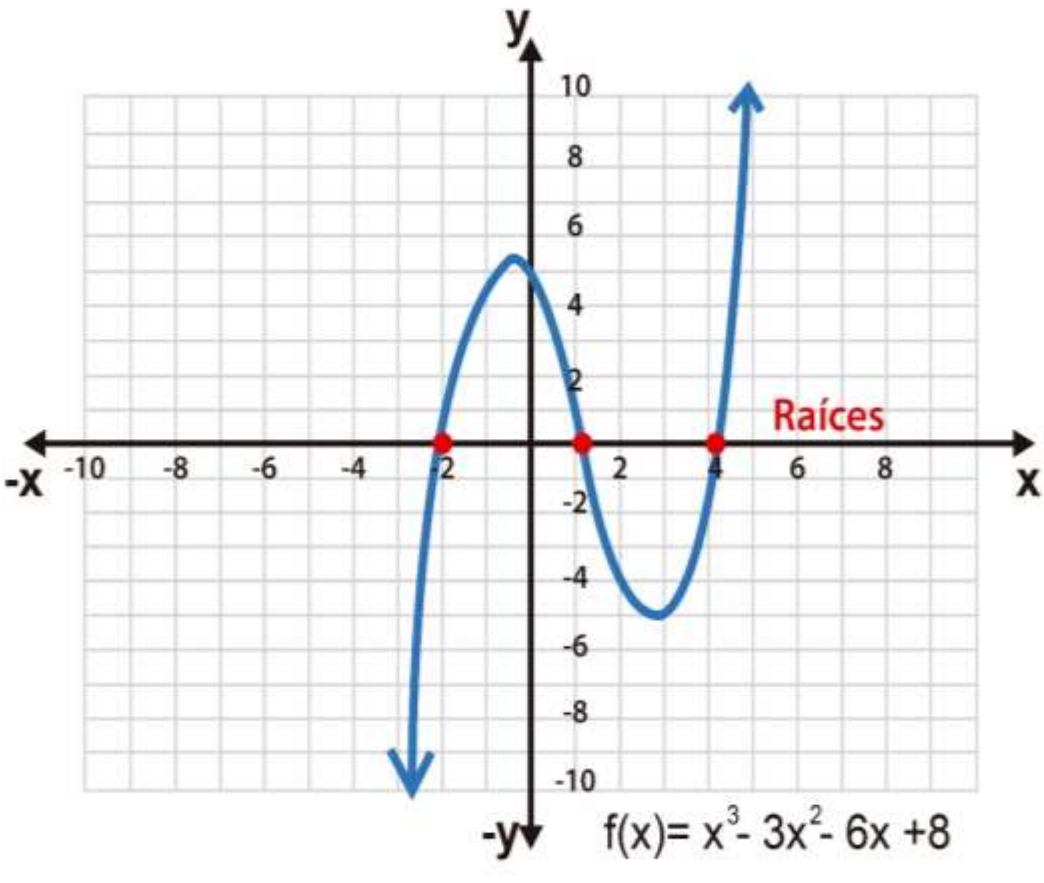

Figura 1. “Gráfica a la solución de una ecuación no lineal con diferentes raíces”, Elaboración propia

En la mayoría de ocasiones hallar esta raíz resulta tedioso o puede que incluso no exista un método mediante cálculos analíticos que lo permitan, en este caso hay que recurrir a otra alternativa como son los métodos numéricos.

Muchos matemáticos y estudiosos en la materia los analizaron dándoles sus características y clasificaciones, pero con el avance de la tecnología se toma muy en cuenta un método numérico respecto con el ámbito computacional. “Desde finales de la década de los cuarenta, la amplia disponibilidad de las computadoras digitales han llevado a una verdadera explosión en el uso y desarrollo de los métodos numéricos” (Chapra, S. C., Canale, R. P., 2011, pág. 4), pero el crecimiento estuvo limitado por el costo de procesamiento y tecnología de esa época . Con el pasar del tiempo hasta ahora, existen varios programas matemáticos especializados, entre estos Matlab (abreviatura de matrix laboratory), el cual es muy usado académica y científicamente.

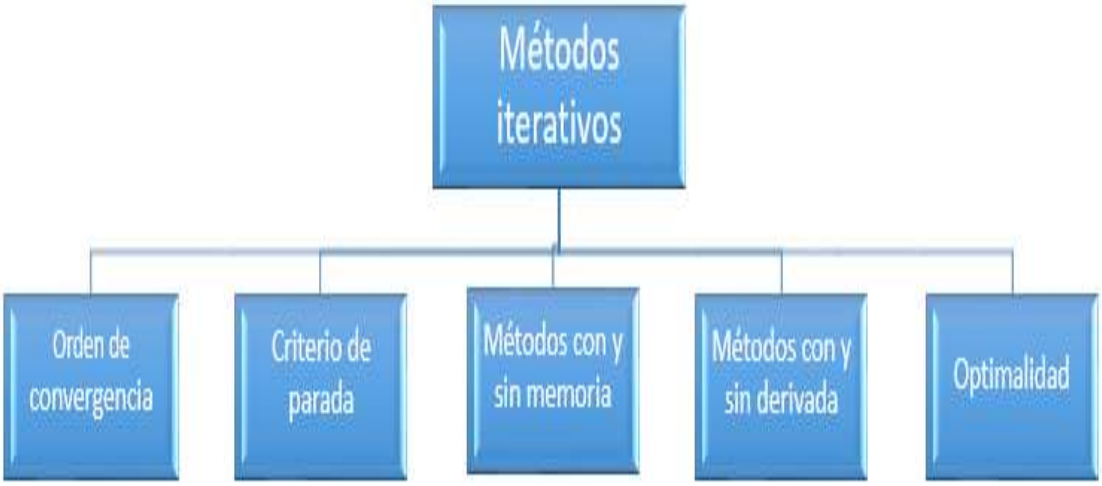

Figura 2. “Clasificación de los métodos iterativos”,

Elaboración propia

Lo que se propone en este trabajo, es un nuevo método usando fundamentos matemáticos y aportando ideas para que el método de Danby pueda ser mejorado en las subclasificaciones de métodos iterativos que se muestran en el esquema de la figura 2, las cuales indicarán si el método creado tendrá éxito.

1.1. Justificación

El concepto teórico-práctico de los métodos iterativos indica que solo obtendremos una aproximación de la solución exacta después de haber utilizado un determinado criterio de parada, el cual tiene condiciones que hacen que el algoritmo del método iterativo pare y no se quede en un ciclo infinito. Al estudiar este tema con más detenimiento me surgió la idea de intentar desarrollar un nuevo método iterativo y poder aplicar los conocimientos aprendidos. Los métodos desarrollados actualmente cuentan con una base matemática de métodos anteriores, el propósito principal con el pasar del tiempo es hacerlos más eficientes y con menos coste de cómputo.

En la mayoría de los métodos iterativos estudiados y que se analizan al hacer los problemas para ecuaciones no lineales tenían una convergencia de orden tres o menor (Newton, Cebichev, Halley, Superhalley, entre otros), en cambio el método de Danby tiene una convergencia de orden cuatro, mayor a los métodos convencionales, por lo que al estudiarlo se pudo observar que al hacer ciertas modificaciones y con el aporte de otras ideas se podría desarrollar un método iterativo, el cual tendría mejores resultados para luego elaborar su algoritmo con las técnicas de programación adecuadas, en el cual podamos obtener la mayor cantidad de información con ayuda de herramientas tecnológicas. Al final deduciendo si esta innovación fue útil.

1.2. Planteamiento del trabajo

En la actualidad es casi imprescindible que se empleen programas computacionales para la investigación científica, a causa de que cada vez se generan problemas más complejos muchas veces excediendo hasta la capacidad del ordenador. En lo que respecta a este trabajo antes de usar el ordenador se necesitará un previo estudio matemático, el cual a partir del desarrollo del método iterativo de Danby se pueda crear un nuevo método iterativo que lo supere en todos los aspectos posibles, dándole el sentido a la investigación realizada y generando otra alternativa para resolver ecuaciones no lineales.

A este método se llamará Danby-Castillo, al que será necesario implementar su algoritmo y mediante comparación de sus resultados determinar si cumple su propósito.

1.3. Estructura de la memoria

En este trabajo se han realizado 6 capítulos, en donde se describirá brevemente cada uno de ellos.

En el capítulo 1 se realizó una introducción describiendo brevemente las ecuaciones no lineales e indicando las clasificaciones de los métodos iterativos, luego se presentó la justificación, el planteamiento del problema por último en la estructura de la memoria se dará un resumen de cada capítulo. En el capítulo 2 se citó trabajos de investigación de otros autores en diferentes artículos, donde se hayan desarrollado un nuevo método iterativo colocando su respectiva referencia bibliográfica para analizar su contenido y compararlo con este trabajo, descartando que haya uno igual. El capítulo 3 contendrá el objetivo concreto, el cual este compuesto por el objetivo general y los objetivos específicos. En el capítulo 4 se colocó los fundamentos teóricos y los conceptos más adecuados acorde a la temática del trabajo. En el capítulo 5 se desarrolló la contribución del trabajo, se lo hizo de manera teórica y práctica con el desarrollo de un ejemplo respecto del tema tratado, el cual explique de manera detallada lo que se quiere aportar. En el capítulo 6 se realizaron las conclusiones también sus ventajas e inconvenientes y a partir de estas, se escribieron las líneas futuras. Por último, en el capítulo 7 se realizó la bibliografía para saber de dónde proviene la información que ayudo a realizar este presente trabajo.

2. Revisión y estado del arte

En este capítulo se analizarán trabajos previos relacionados con esta investigación que se refiere a métodos numéricos para hallar la raíz de una ecuación no lineal de la forma $f(x) = 0$. En este campo hay muchos estudios, pero se tendrán en cuenta los que consideremos más importantes.

- Ezquerro, J. A., Gutiérrez, J. M., Hernández, M. A., & Salanova, A. (2001). *El método de Halley: Posiblemente, el método más redescubierto del mundo*. Servicio de Publicaciones, Universidad de la Rioja: En esta publicación se desarrollan varios métodos iterativos, pero analizaremos dos métodos, el método de Newton y el de Halley. El método de Newton quizá el más famoso, surgió con un primer planteamiento de Newton, en el cual se explicaba que al resolver una ecuación no lineal, primero se da un valor por tanteo y se le suma un épsilon, al cual se lo evalúa en la función quedando $f(x_0 + \varepsilon) = 0$, de la ecuación se ignoran los términos ε^2 o con mayores potencias por ser épsilon un valor muy pequeño. Al final se despeja el valor de épsilon y simplemente se le suma al valor inicial, dando la primera iteración y así se continua hasta cumplir un cierto criterio de parada. Pero para el método iterativo actual de Newton, la incorporación del cálculo diferencial se debe a Thomas Simpson, quien hizo al método de Newton tal y como lo conocemos actualmente con orden de convergencia cuadrática. En el método de Halley también hay varias maneras de llegar a su modelo de ecuación iterativa, pero se explicará por el desarrollo de Taylor. Se empieza por una aproximación x_0 de la solución de una ecuación $f(x) = 0$, se cambia por una aproximación de la forma $x_0 + \varepsilon$. Evaluando en la serie de Taylor y en el tercer término despejamos épsilon, como tenemos dos respuestas una con signo negativo y otra con signo positivo, escogemos la respuesta del signo positivo. Esta ecuación queda con raíces cuadradas, para evitar esto se usa correctores y queda tal como en la actualidad con orden de convergencia cúbico.

- In Yun, B. (2011). *A quadratically convergent iterative method for nonlinear equations*. J. Korean Math. Soc., 48(3), 487-497. doi:10.4134/JKMS.2011.48.3.487: En este trabajo se propone un método iterativo, el cual no requiere derivadas y es de convergencia cuadrática. El desarrollo de este método empieza por asumir una función continua f, que tiene un único cero en el intervalo $[a, b]$ con $f(a).f(b) < 0$

y el valor inicial x_k elemento de (a, b) una aproximación a la raíz p de la ecuación $f(x) = 0$. Al dejar establecido $a_k = x_k - h_k$ y $b_k = x_k + h_k$ entonces p estará incluida en el subintervalo $[a_k, b_k]$ o $([b_k, a_k])$ de [a, b] para algún $h_k \neq 0$. En donde L(p; x) y L(x_k, x) son dos funciones lineales por parte. L(p; x) y L(x_k, x) interpolan a f(x) en los puntos $x = a_k, p, b_k$ y $x = a_k, x_k, b_k$ respectivamente. Luego se definen dos integrales para las dos funciones respectivas, luego reemplazamos p por x_{k+1} en la integral $L_k(p)$ y resolviendo la ecuación $I_k(x_{k+1}) = J_k$, luego se desarrolla la ecuación final y para el valor inicial x_0 se sugiere $x_0 = ((a + b) / 2)$.

- Abu, I., & Alshaikh. (2005). *A New Iterative Method for Solving Nonlinear Equations*. Proceedings of World Academy of Science, Engeneering and Technology, 5: En el desarrollo de este nuevo método necesitaremos dos valores, en los cuales no necesariamente la raíz este dentro de su intervalo pero que al menos quede cerca, luego asumimos que f(x), f ’(x) y f ”(x) son continuos y cercanos a la raíz α, después el gráfico f(x) y – f(x) intersecta a x en las abscisas en el punto (α, 0). Luego asumimos que la aproximación inicial x_0 y x_1 son cercanos a α y $x_0 \neq x_1$. Después los puntos $(x_0, f(x_0))$ y $(x_1, f(x_1))$, están entre los que forman la curva y = f(x) cerca del punto (α, 0). La información respecto a la naturaleza de f(x) y – f(x) puede ser usada para desarrollar un algoritmo que producirá una secuencia $\{x_{k+1}\}$ que converge a la raíz α. Debemos introducir el algoritmo a la secuencia gráficamente y luego para hacer pruebas de error y convergencia se usa el desarrollo de Taylor. De las gráficas f(x) y – f(x) tendremos pendientes m_1 y m_2, de la línea L_1 definida por el punto $(x_0, 0)$ y pasa a través del punto $(x_1, -f(x_1))$, donde L_2 es la línea dada por el punto $(x_1, 0)$ que pasa a través del punto $(x_2, -f(x_2))$. Para completar la ecuación del método debemos aplicar diferencias divididas hacia atrás y hacia adelante en la ecuación de x_1. Luego hacemos sustituciones, reemplazamos hallando x_2 y x_3 y con esto, se puede obtener la ecuación final.

- Calderón, G. (2007). *Métodos iterativos para resolver ecuaciones no lineales*. Revista Notas de Matemática, 3(250), 33-44.: En este artículo se presenta un nuevo método predictor-corrector para resolver ecuaciones no lineales usando los métodos de Bisección y Müller (BM). El método es desarrollado especialmente para funciones de tipo exponencial o en los casos donde se tiene un cambio de concavidad dentro del intervalo [a, b]. Para el desarrollo se usa el método de bisección como predictor estableciendo un intervalo [a, b] tal que $f(a).f(b) < 0$ por lo que sabemos que existe, al menos una raíz real. Con ayuda del intervalo hallamos c y con esto se halla los coeficientes del polinomio de segundo grado a_0, a_1, a_2 para reemplazar en el método de Müller, que usaremos como corrector para encontrar la raíz con un determinado criterio de parada.

- Biazar, J., & Ghanbari, B. (2008). *A New Computational Approach for Nonlinear Equations*. International Mathematical Forum, 3(20), 955-960.: Este nuevo método iterativo está hecho para resolver ecuaciones no lineales y está basado en el método de Newton y series de potencia. El orden de convergencia es cúbico. El desarrollo del método comienza al tener una ecuación no lineal $f(x) = 0$ y al escribirla de forma truncada en la serie de Taylor alrededor de una aproximación inicial a la solución de la ecuación, se desarrolla esta serie hasta el término tres suponiendo que x_0 diferente de 0 y se busca el valor h tal que $f(x_0 + h) = 0$ y nos queda la segunda derivada de la ecuación determinada por h. Luego usando propuestas matemáticas y artificios llegamos la ecuación iterativa $x_{n+1} = x_n - h$, al final se demuestra la convergencia del análisis del nuevo método.

- Ahmad, N., Rafiq, N., & Akram, S. (2009). *An Efficient Three-Step Iterative Method for Non-Linear Equations.* International Journal of Mathematical Analysis, 3(40), 1989-1996.: En esta investigación se desarrolla un nuevo método iterativo de tres pasos para resolver ecuaciones no lineales. En este trabajo se demuestra que el método tiene un orden de convergencia 8. Algunos ejemplos numéricos dan prueba de la eficiencia y el rendimiento del método. El desarrollo del método inicia al tener la ecuación no lineal $f(x) = 0$, a la cual damos un valor aproximado a la raíz y lo evaluamos en el método de Newton que usamos como primer predictor. En el segundo paso, la respuesta dada en el primer predictor la evaluamos en el segundo predictor que es una modificación del método de Mamta en (Mamta, 2005), por último evaluamos en un método corrector, el cual es una composición de los dos métodos iterativos anteriores.

En todos los trabajos de investigación y artículos buscados anteriormente ninguno describe el método a desarrollar que se hará en este trabajo.

3. Objetivos concretos

Realizadas la justificación y el planteamiento del problema, nos serán útil para desarrollar el objetivo general y objetivos específicos que se presentarán a continuación.

3.1 Objetivo general

Desarrollar un nuevo método iterativo con mayor orden de convergencia e índices de eficiencia en base a las modificaciones matemáticas realizadas al método iterativo de Danby y al análisis de las simulaciones efectuadas.

3.2 Objetivos específicos

- Explicar de manera teórica las variantes que se utilizaron en el desarrollo matemático del método propuesto.
- Demostrar analíticamente el orden de convergencia del método propuesto.
- Analizar las subclasificaciones estudiadas de los métodos iterativos, con los resultados numéricos obtenidos con el ejemplo práctico, tanto al método de Danby como al propuesto.
- Describir del método propuesto, la realización de su algoritmo en Matlab R2018a y efectuar los cálculos y gráficas requeridos en el software para este trabajo.

4. Marco teórico

A continuación, se describe los fundamentos y conceptos que conformarán el marco teórico en el que se sustenta el presente trabajo. Cabe mencionar que dará un mayor entendimiento en general del tema tratado.

4.1 El inicio hacia el uso de métodos iterativos

Se realizará este escrito basándonos en (Stewart I. n., 2008)**,** (Mario Dalcín, 2011)**,** (Muñoz, 2011)**,** (Sánchez , 2011)**,** (Martin casaderrey, 2000)**,** (Corbalán, 2010) **y** (Chapra, S. C., Canale, R. P., 2011)**.**

En la resolución de raíces de polinomios grado 1 y 2 los egipcios y los babilonios fueron los pioneros aproximadamente por el 2000 a. C. Los egipcios dejaron en sus papiros multitud de problemas matemáticos resueltos, casi todos eran de polinomios grado 1 y de tipo aritméticos que se relacionaban con cosas de la vida cotidiana, encontrándose algunos que se pueden clasificar como algebraicos. Los Babilonios en cambio les dedicaron mayor atención a los polinomios de grado 2 y a los sistemas de ecuaciones que a las ecuaciones lineales. Con el pasar del tiempo los matemáticos mejoraron sus fundamentos y desarrollaron fórmulas específicas para resolver estas ecuaciones polinómicas, como la conocida fórmula cuadrática que la desarrolla finalmente como se observa en la actualidad el matemático Francisco Vieta a mediados del siglo XV. Introduciendo letras para expresar de forma general la ecuación, las consonantes para los datos y las vocales para las incógnitas, esto lo publicó en el libro Canon Mathematicus en 1570. El desarrollo de la ecuación para resolver el polinomio de grado 3 la hizo el italiano Nicolo Tartaglia en el año de 1535 en uno de sus escritos en Venecia y del polinomio de grado 4 fue también un italiano llamado Ludovico Ferrari y estas dos soluciones fueron publicadas en el libro *Ars Magna* en 1545.

Pero para todos los polinomios no se pueden desarrollar ecuaciones para hallar sus raíces en término de sus coeficientes, esto lo descubrió el matemático noruego Niel Henrik Abel, en este caso para polinomios de grado $n \geq 5$ y en 1824 fue publicada de manera difícil y no tan entendible. Después la publicó de manera más elaborada en el diario de Crelle. El matemático francés Evariste Galois en 1832 no solo confirmó lo postulado por su colega Niel Henrik Abel, sino que profundizó más, al caracterizar las ecuaciones que sí tenían solución y las nombró el "grupo de la ecuación", con esta idea se determinaba si la ecuación era resoluble o no.

Al tener limitaciones de no poder resolver todas las ecuaciones polinómicas y la gran mayoría de otros tipos de problemas matemáticos por métodos directos, se originaron los métodos iterativos o indirectos. Estos métodos se dividen en *métodos cerrados* porque se necesitan dos valores iniciales para hallar la raíz, en donde estos valores deben estar a ambos lados de la raíz para converger a la respuesta correcta. Entre los métodos que entran en esta clasificación están bisección o posición falsa.

Los *métodos abiertos* son los que requieren un solo valor de inicio o que empiecen con dos valores, pero que no necesariamente encierren a la raíz. Los métodos en esta clasificación son método de punto fijo, Secante, Steffensen entre otros.

4.2 Métodos iterativos de punto fijo

El método a realizarse en este trabajo entra en esta clasificación y consiste en resolver ecuaciones que parten de la forma $f(x) = 0$ y transformarla en una ecuación $x = g(x)$ (siendo $g(x)$ el operador), en la cual la raíz o raíces de x serán los puntos fijos que la satisfagan. En la actualidad la mayoría de métodos iterativos son de punto fijo, es decir evaluamos una aproximación inicial en el operador del método iterativo correspondiente en una secuencia de iteraciones hasta alcanzar a cierto criterio de parada. El criterio de parada se satisface cuando la aproximación es lo bastante buena.

Teorema 4.1 (punto fijo). Si g es una función continua en $[a, b]$ y su operador $g(x) \in [a, b]$, entonces tiene al menos un punto fijo en $[a, b]$. Si se agrega que g es derivable y que $g'(x) \in [a, b]$ para todo $x \in [a, b]$ y $|g'(x)| \leq k < 1$ para todo $x \in [a, b]$, k constante, entonces g tiene un único punto fijo $x \in [a, b]$ (Burden & Faires, 1998).

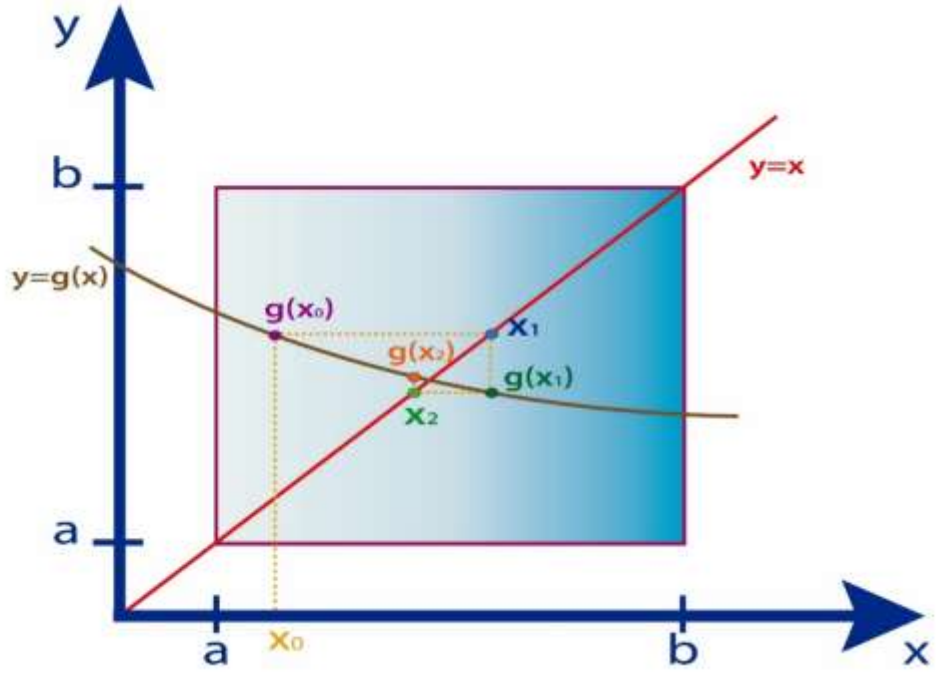

Figura 3. “Modelo general del método iterativo punto fijo”,

Elaboración propia

entonces la sucesión x_{n+1} se halla mediante la siguiente ecuación de iteración,

$$x_{n+1} = g(x_n), n = 1, 2, 3...$$

Al encontrar la raíz x^* se cumplirá por la teoría que $x^* = g(x^*)$, en donde será un punto fijo de la función g también conocida como función de punto fijo. Hay muchas formas de obtener un método iterativo de punto fijo, a consecuencia de cambiar la manera de definir su función g. Cuando la función g depende solo del resultado de la iteración anterior, al método iterativo de punto fijo se lo conoce como *métodos sin memoria* $x_{n+1} = g(x_n)$ y cuando la función g depende no solo del resultado de la iteración anterior sino de anteriores, el método iterativo es conocido como *métodos con memoria* $x_{n+1} = g(x_n, x_{n-1}, \dots, x_{n-i})$, i $\leq$ n.

Los métodos iterativos de punto fijo se clasifican por sus evaluaciones funcionales en: métodos de un punto o métodos multipunto. Cuando la iteración (n+1)-ésima solo depende de la evaluación n-ésima se la conoce como *métodos de un punto*, $x_{n+1} = \emptyset(x_n)$.

En caso que la iteración (n+1)-ésima dependa de otras evaluaciones funcionales, es decir, aparte del iterado n-ésimo haya otras evaluaciones intermedias, se lo conoce como *métodos multipunto* o *predictor-corrector*

$$y_{n+1} = \pitchfork(x_n)$$
$$x_{n+1} = \varnothing(x_n, y_n)$$

Otra clasificación de los métodos de punto fijo son los métodos con derivadas y sin derivadas. El método iterativo más conocido es el método de Newton, que se presenta a continuación,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4.1}$$

observando las clasificaciones anteriores, este método es sin memoria, de un punto y con derivada.

Para los métodos iterativos aplicados a ecuaciones no lineales se necesita un criterio de parada que defina el momento que se detendrá el algoritmo que se está ejecutando, caso contrario se realizarían iteraciones infinitas pero para abordar este tema primero debemos definir basándonos en (Nieves & Dominguez, 2014), tres tipos de errores que son: el *error absoluto*, el cual calcula el número de decimales correctos

$$e_a = |x_{n+1} - x_n| \tag{4.2}$$

otro es el *error relativo*, que calcula el número de cifras significativas

$$e_r = \frac{|x_{n+1} - x_n|}{x_{n+1}} \tag{4.3}$$

por último, el *error relativo porcentual* aproximado, que es el error relativo en términos porcentuales

$$e_{rp} = \frac{|x_{n+1} - x_n|}{x_{n+1}} * 100\% \tag{4.4}$$

El proceso más usual para utilizar el criterio de parada es utilizar un parámetro llamado *tolerancia* (Tol), el cual es un valor muy pequeño y se escoge dependiendo la precisión que se necesite. Parándose el proceso al cumplirse una de las siguientes condiciones:

1. $|x_{n+1} - x_n| < Tol$

2. $\frac{|x_{n+1} - x_n|}{x_{n+1}} < Tol$

4.2.1 Convergencia

La velocidad con la cual una sucesión finita $\{x_n\}_{n\geq 0}$ converge a la solución es el denominado *orden de convergencia*. De cualquier proceso iterativo se distinguen tres tipos de convergencia. La primera es la *convergencia local*, que supone que existe un conjunto de valores próximos a la solución y convergen a esta. La segunda *convergencia semilocal* señala que para un conjunto de valores iniciales se puede asegurar su convergencia a la solución **y** la *convergencia global*, la cual indica que para un conjunto grande de aproximaciones iniciales se converge a la solución estimada.

La convergencia que se usará en este trabajo, será la local y se define en (Burden & Faires, 1998, págs. 78-79): supongamos que $\{x_n\}_{n=0}^{\infty}$ es una sucesión que converge a x^*, con $x_n \neq x^*$ para toda n. Si existen constantes positivas c y p con:

$$\lim_{n\to\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^p} = \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^p} = c$$

entonces $\{x_n\}_{n=0}^{\infty}$ converge a x con orden p y una constante de error asintótico c. Se dice que un método iterativo de la forma $x_n = g(x_{n-1})$ es de orden p, si la sucesión $\{x_n\}_{n=0}^{\infty}$ converge a la solución $x^* = g(x^*)$ con orden p.

si $p = 1$ y $0 < c < 1$: convergencia lineal

si $p > 1$ y $c > 0$: convergencia de orden p no lineal (cuadrática, cúbica, cuártica,..)

Para hallar el orden de convergencia p de un método iterativo debe satisfacer la siguiente ecuación del error

$$e_{n+1} = c\, e_n^p + O(e_n^{p+1}) \tag{4.5}$$

para la cual, es importante conocer previamente la notación $e_n = x_n - x^*$ y el error $O_n = O(e_n^p)$, en donde para encontrar la ecuación del error de un método iterativo, es muy utilizado el desarrollo de la serie de Taylor. Hay otra técnica para encontrar el orden de convergencia y la ecuación del error, esta técnica es a partir del teorema de Schröder, el cual se utilizará en este trabajo y se hará su demostración argumentándonos en (Kincaid, 1991), a continuación.

Teorema 4.2 (de Schröder). Al tener un método iterativo $x_{n+1} = g(x_n)$, en el que g es q veces diferenciable en el entorno de x^*, un punto fijo de g, $x^* = g(x^*)$. Entonces la sucesión $\{x_n\}$ converge a x^* con orden de convergencia q si:

$$\frac{d^j\, g(x^*)}{dx^j} = 0,\ j = 1,2,\ldots,\ q-1,\ \frac{d^q\, g(x^*)}{dx^q} \neq 0.$$

Demostración: Al desarrollar $x_{n+1} = g(x_n)$, en series de Taylor queda de la siguiente manera

$$x_{n+1} = g(x_n) = x^* + g'(x^*)\,(x_n - x^*) + \ldots + \frac{1}{q!}\frac{d^q\, g(x^*)}{dx^q}(x_n - x^*)^q,\ x \in (x_n, x^*)$$

luego, usando la condición del enunciado sigue

$$x_{n+1} - x^* = \frac{1}{q!}\frac{d^q\, g(x^*)}{dx^q}(x_n - x^*)^q,$$

con lo que,

$$\lim_{n\to\infty}\frac{|x_{n+1} - x^*|}{|x_n - x^*|^q} = \frac{1}{q!}\frac{d^q\, g(x^*)}{dx^q} \neq 0$$

por tanto, el método tiene orden q.

La siguiente fórmula para funciones suficientemente derivables, es para definir la constante del error asintótico de los métodos iterativos que se estudiarán más adelante y es la siguiente:

$$a_j = \frac{f^{\,j}(x^*)}{j!\, f'(x^*)} \tag{4.6}$$

Algo importante que se debe mencionar con la convergencia respecto a métodos de un punto basándonos en (A. Cordero and J. R. Torregrosa, 2007), es que para diseñar un método iterativo de un punto de orden p, su expresión iterativa debe contener derivadas al menos hasta orden p – 1.

Cuando se usa un método predictor y otro corrector una de las maneras de hallar su orden de convergencia es por el teorema en (Traub, H.T. Kung. J.F, 1974), a continuación:

Teorema 4.3 Sean $g_1(x)$ y $g_2(x)$ dos funciones de punto fijo para $f(x) = 0$ consideremos los esquemas iterativos $x_{n+1} = g_1(x_{n+1})$ y $x_{n+1} = g_2(x_{n+1})$ de ordenes p_1 y p_2 respectivamente. Entonces el orden de convergencia del método iterativo asociado a la función de punto fijo $g(x) = g_2\ (g_1(x))$ es $p_1\ .\ p_2$.

El nivel de garantía de convergencia que ofrece un método iterativo se le conoce como estabilidad, se menciona esto, a causa de que hay casos que los métodos iterativos divergen.

4.2.2 Índices de eficiencia

Al hacerse parte esencial las ciencias de la computación de los métodos numéricos, no solo se hacen clasificaciones desde el análisis matemático sino también respecto al computacional. En índices de eficiencia habrá la clasificación tanto desde el enfoque matemático como del coste de cómputo.

Entre estas tenemos el *índice de eficiencia* propuesto por Ostrowski en (Ostrowki, 1964), que está dado por su orden de convergencia con la letra p, para $p > 1$ y el número de evaluaciones funcionales con la letra d

$$IE = p^{1/d} \tag{4.7}$$

también tenemos la *eficiencia informacional* hecho por Traub en (J.F Traub, 1964) donde p es el orden de convergencia y d es el número de evaluaciones funcionales

$$I = \frac{p}{d} \tag{4.8}$$

El *índice de eficiencia computacional* en (A. Cordero and J. R. Torregrosa, 2007), nos indica mediante el número de evaluaciones funcionales más el número de evaluaciones producto/cociente la eficiencia de un método, donde d es el número de evaluaciones funcionales por iteración y op es el número de evaluaciones (producto/cociente) por iteración

$$\mathrm{IC} = \mathrm{p}^{1/(\mathrm{d}+\mathrm{op})} \tag{4.9}$$

Para estimar el orden de convergencia, un método más reciente es el realizado por Weerakoon y Fernando en (S. Weerakoon and T. G. I. Fernando A., 2000), llamado *orden de convergencia computacional* (COC)

$$\mathrm{COC} = \frac{\ln(|x_{n+1} - x^*| \,/\, |x_n - x^*|)}{\ln(|x_n - x^*| \,/\, |x_{n-1} - x^*|)} \tag{4.10}$$

en donde, x^* es la raíz de la ecuación usando tres iteraciones consecutivas x_{n+1}, x_n y x_{n-1} pero en la práctica no tenemos la solución x^*, por lo que Cordero y Torregrosa innovaron un nuevo método para estimar la convergencia llamado *orden de convergencia computacional aproximado* (ACOC), en (A. Cordero and J. R. Torregrosa, 2007), el cual no necesita x^* pero usa cuatro iteraciones consecutivas x_{n+1}, x_n, x_{n-1} y x_{n-2}

$$\mathrm{ACOC} = \frac{\ln(|x_{n+1} - x_n| \,/ |x_n - x_{n-1}|)}{\ln(|x_n - x_{n-1}| \,/ |x_{n+1} - x_{n-2}|)} \tag{4.11}$$

Para verificar y comparar la efectividad de un método numérico es necesario caracterizarlo y esto es lo que hace la *tasa de convergencia*

$$\frac{|x_{n+1} - x_n|}{|x_n - x_{n-1}|^p} \approx \text{cte}, \ \forall \ n \geq n_o \tag{4.12}$$

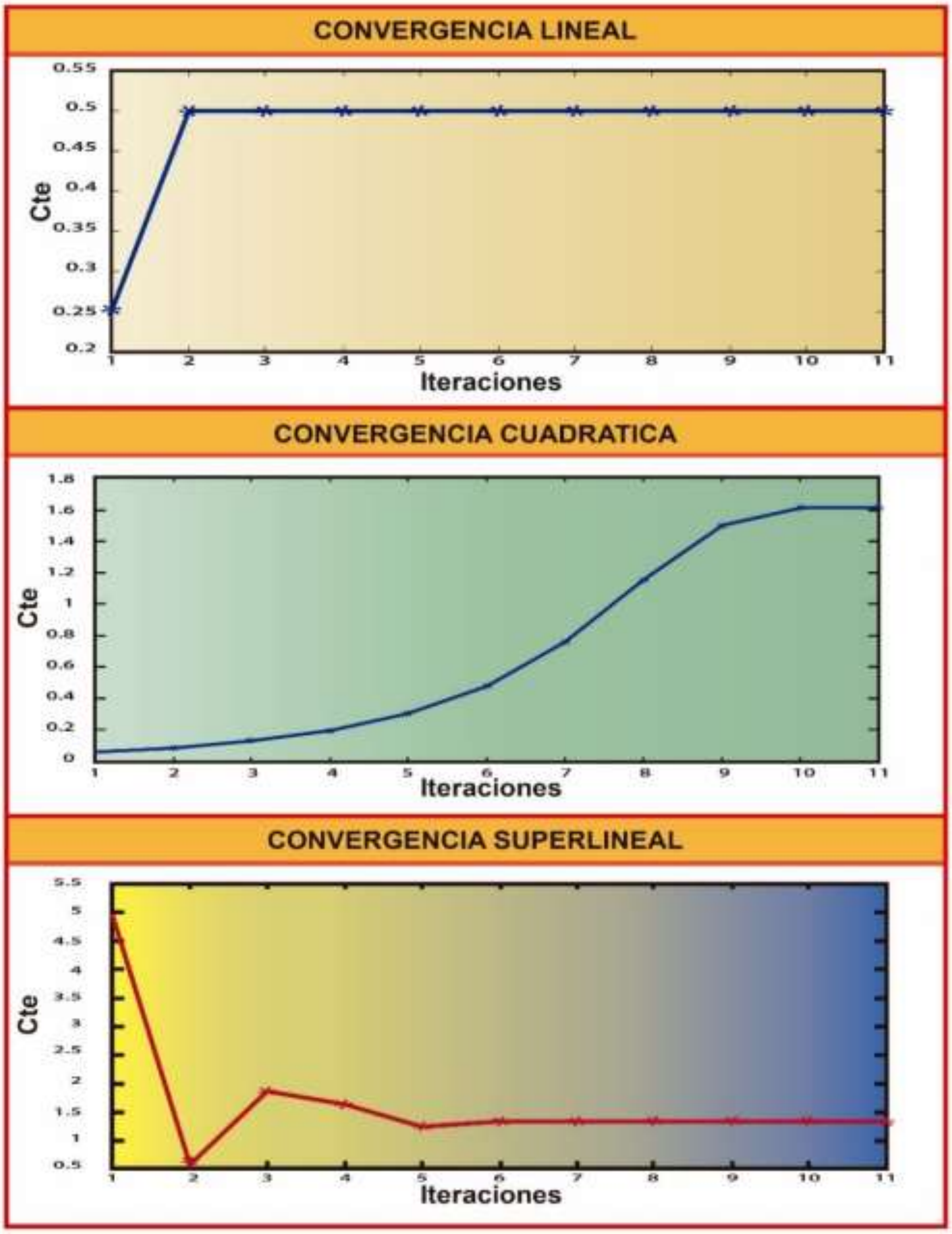

Figura 4. "Muestras de gráficos de diferentes tasas de convergencia",

Elaboración propia

La Conjetura de Kung y Traub en (H.T. Kung and J.F. Traub), la cual indica que el orden de convergencia de un método iterativo sin memoria, con d evaluaciones funcionales por iteración no puede superar la cota de:

$$2^{d-1}$$

de donde, d es el número de evaluaciones funcionales, cuando el método logra esta cota, se le llama óptimo y cuando no, se le llama no óptimo.

Hemos hecho un estudio de soluciones por métodos numéricos para una sola ecuación, pero este campo se extiende para más de una, generándose los sistemas de ecuaciones por lo que se puede suponer que es mucho más complicada su resolución. En esta investigación se realizará el estudio para una sola ecuación, pero se quiere dejar una pequeña introducción en cuanto a la resolución de sistemas, con su estudio en (Cabezas, 2018).

Para encontrar una solución real β de un sistema de ecuaciones no lineales,

$$\left.\begin{array}{l} f_1\ (x_1, x_2, \ldots.., x_n\ ) = 0 \\ f_2\ (x_1, x_2, \ldots.., x_n\ ) = 0 \\ f_n\ (x_1, x_2, \ldots.., x_n\ ) = 0 \end{array}\right\} \qquad F(X) = 0, \quad F: D \subseteq R^n \longrightarrow R^n$$

donde, $R^n \longrightarrow R^n$, $i = 1, 2 \ldots. n$ son las funciones coordenadas.

Para aproximar su solución se utilizan métodos iterativos de punto fijo descritos por una función G: $R^n \longrightarrow R^n$,

$$x^{m+1} = G^m, \quad m = 0, 1, 2 \ldots$$

Las subclasificaciónes estudiadas de los métodos iterativos para ecuaciones no lineales como orden de convergencia, índice de eficiencia, índice de eficiencia computacional, criterio de parada entre otros, se están adaptando con continuo estudio para sistemas. Para ecuaciones no lineales dependiendo el método iterativo a usar, se utiliza la derivada, pero en sistemas se utiliza su analogía que es la jacobiana.

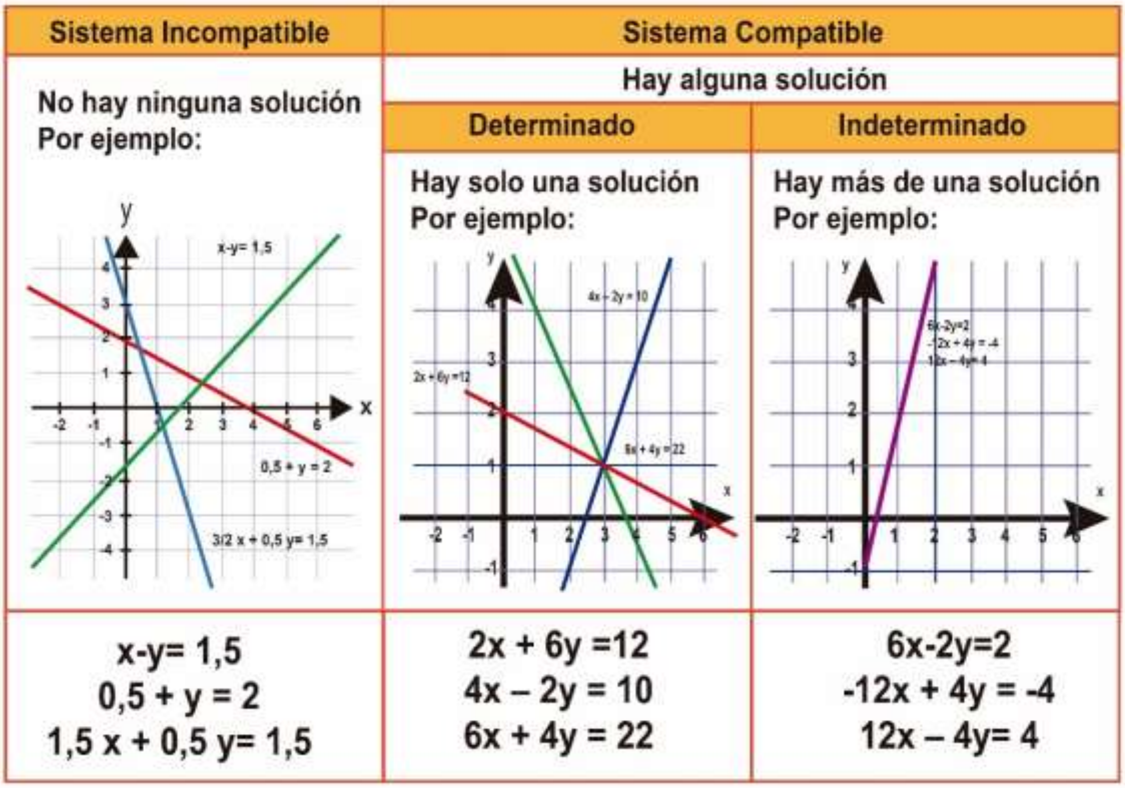

Figura 5. "Representación gráfica de tipos de soluciones a sistemas de ecuaciones",

Elaboración propia

4.3 Biografía de Danby

Para empezar a explicar la aplicación del método iterativo de Danby, se dará a conocer una breve información de lo más relevante de la biografía de su autor en (Alan Hirshfeld , 2017), (Danby, 1988) y el motivo por lo que lo realizó. J. M. A. Danby nació en 1929 en Londres y falleció el 2009 en Pennsylvania. Obtuvo un doctorado de Astronomía en la Universidad de Manchester, en 1957 fue profesor en la Universidad de Minnesota y cuatro años después entró al departamento de Astronomía de la Universidad de Yale. En 1962 publicó su clásico libro llamado *Fundamentos de la mecánica celeste* y en 1967 se incorporó como profesor de matemáticas en la Universidad Estatal de Carolina del Norte, en la cual estuvo hasta jubilarse en 1998.

El método de Danby también se conoce como el método de Danby-Burkardt y este es otra opción a los métodos convencionales (Newton, Steffensen, Halley, ect) para resolver la ecuación de Kepler,

$$f(E) = 0, \text{ con } f(E) = E - e \text{ sen } E - M \tag{4.13}$$

de esta ecuación M es valor de la anomalía media y e el valor de la excentricidad de una órbita elíptica.

En lo que diferencia los métodos convencionales al método de Danby es que este fue hecho especialmente para resolver la ecuación de Kepler, pero en general resuelve ecuaciones de la forma f(x) = 0.

4.3.1 Realización de su método

Este autor aporto con una nueva idea para mejorar los métodos tradicionales y aumentar el orden de convergencia, su proceso en (J. M. Danby, 1983), será explicado a continuación:

Primero se obtiene una familia de métodos iterativos con orden k para resolver una ecuación no lineal f(x) = 0. Obteniéndola de esta manera, sea x^* una solución de f(x) = 0, x_n una aproximación de x^* y $\epsilon_n = x^* - x_n$ el error cometido, entonces:

$$0 = f(x^*) = f(x_n + \epsilon_n) = f(x_n) + f'(x_n)\,\epsilon_n + \ldots + \frac{f^k x_n}{k!}\,\epsilon_n^k + \ldots$$

supongamos, que el desarrollo de Taylor truncado en el término k-ésimo tiene una raíz, que denotaremos δ_n, es decir,

$$0 = f(x_n) + f'(x_n)\,\delta_n + \ldots + \frac{f^k x_n}{k!}\,\delta_n^k \tag{4.14}$$

de esta ecuación δ_n es la solución y comprueba que converge a x^* con orden k + 1, entonces Danby define el proceso iterativo

$$x_{n+1} = x_n + \delta_n$$

La idea de Danby consiste en dar un paso más a este proceso. En concreto para k = 3 la ecuación puede escribirse como:

$$\delta_n = -\frac{f(x_n)}{f'(x_n) + \frac{1}{2} f''(x_n)\delta_n + \frac{1}{6} f'''(x_n)\delta_n^2}$$

la clave ahora es elegir las aproximaciones δ_n de la parte derecha de la ecuación. Aquí se propone una elección en dos pasos

$$\delta_{n1} = -\frac{f(x_n)}{f'(x_n)}$$

$$\delta_{n2} = -\frac{f(x_n)}{f'(x_n) + \frac{1}{2} f''(x_n)\delta_{n1}}$$

$$\delta_n = -\frac{f(x_n)}{f'(x_n) + \frac{1}{2} f''(x_n)\delta_{n2} + \frac{1}{6} f'''(x_n)\delta_{n2}^2}$$

el método a realizar es:

$$x_{n+1} = x_n + \delta_n$$

teniendo en cuenta las notaciones conocidas del grado de convexidad logarítmica

$$L_{f(x)} = \frac{f(x)\, f''(x)}{f'(x)^2} \tag{4.15}$$

y

$$L_{f'(x)} = \frac{f'(x)\, f'''(x)}{f''(x)^2} \tag{4.16}$$

por último, propone de su autoría la siguiente ecuación iterativa

$$x_{n+1} = x_n - \frac{3}{2}\frac{(2 - L_f(x_n))^2}{6 - 9L_f(x_n) + 3L_f(x_n)^2 + L_f(x_n)^2\, L_{f'}(x_n)}\frac{f(x_n)}{f'(x_n)} \tag{4.17}$$

Esta es la ecuación que propone Danby y la que usaremos para cálculos y algoritmos posteriores, pero esta ecuación también puede tener una variación y es despreciando a los términos que contengan a $f^4\,(x_n)$, en donde queda de la forma

$$x_{n+1} = x_n - \left(1 + \frac{1}{2} L_f(x_n) + \frac{1}{2} L_f(x_n)^2 - \frac{1}{6} L_f(x_n)^2 L_{f'}(x_n)\right)\frac{f(x_n)}{f'(x_n)}$$

Danby también propone métodos iterativos de orden de convergencia 5 para acelerar el proceso de su método observado anteriormente.

4.3.2 Orden de convergencia local del método

Teorema 4.4 Sea f: I ⟶ R una función de variable real definida en un entorno I de x^*, una raíz simple de f(x) = 0, ($f(x^*) = 0$ y $f'(x^*) \neq 0$). Supongamos que f es cuatro veces derivable con continuidad en I. Entonces el método de Danby definido por $x_{n+1} = Db_f(x_n)$

$$x_{n+1} = x_n - \frac{3}{2}\frac{(2 - L_f(x_n))^2}{6 - 9L_f(x_n) + 3L_f(x_n)^2 + L_f(x_n)^2\, L_{f'}(x_n)}\frac{f(x_n)}{f'(x_n)}$$

tiene al menos, convergencia local de orden 4.

Demostración: Se utiliza el teorema de Schröder (ver teorema 4.2), de donde el cálculo de derivadas Db_f nos permite deducir lo siguiente

$$Db_f(x^*) = x^*, Db_f^{(i)}(x^*) = 0, i = 1, 2, 3$$

$$Db_f^{(4)}(x^*) = \frac{3f^{(2)}(x^*)^3 - 2f^{(1)}(x^*)\, f^{(3)}(x^*)\, f^{(2)}(x^*) + f^{(1)}(x^*)^2\ \ f^{(4)}(x^*)}{f'(x^*)^3} \neq 0$$

por ende,

$$e_{n+1} = x_{n+1} - x^* = Db_f(x_n) - x^* = \frac{Db_f^{(4)}(x^*)}{4!}\ (x_n - x^*)^4 + O_5,$$

de donde, usando la notación (4.6), se deduce

$$\lim_{n\to\infty}\frac{e_{n+1}}{e_n^4} = \frac{Db_f^{(4)}(x^*)}{4!} = a_2^3 - a_2a_3 + a_4$$

y su constante del error asintótico es $a_2^3 - a_2a_3 + a_4$ cumpliendo la siguiente ecuación del error:

$$e_{n+1} = (a_2^3 - a_2a_3 + a_4)\, e_n^4 + O_5$$

por lo tanto, es un método con orden de convergencia local igual a 4.

5. Desarrollo de la contribución

5.1. Realización del método propuesto

El nuevo método propuesto “Danby-Castillo” es una modificación del método de Danby. Se ha mencionado anteriormente que el método de Danby fue realizado para resolver en la práctica una ecuación $f(x) = 0$, pero este método tiene varios aspectos negativos como una alta cantidad de operaciones matemáticas y por ende un alto coste computacional, la cantidad alta de derivadas que es 3, una no muy buena eficiencia entre otras.

El método propuesto tendrá un método predictor que será el de Newton, el cual aumentará su orden de convergencia y optimalidad y tendrá otro corrector que será un método iterativo de la familia de iteraciones de Danby, diferente del método iterativo escogido para su ecuación original.

Es importante saber el desarrollo para llegar al método predictor (Newton). Hay varias maneras de realizarlo en este caso se lo hará por series de Taylor basándonos en (D. Kincaid, 1991), a continuación.

Si es x_0 una aproximación de x^*, la raíz de $f(x) = 0$. Se busca un término corrector h de forma que x + h sea la raíz buscada, es decir, $f(x_0 + h) = f(x^*) = 0$.

si desarrollamos la serie de Taylor $f(x_0 + h)$ obtenemos:

$$0 = f(x_0 + h) = f(x_0) + h f'(x_0) + \frac{h^2}{2!} f''(x_0) + \ldots\ldots$$

En donde, solo tomaremos hasta el segundo término de la serie de Taylor

$$f(x_0) + h f'(x_0) = 0,$$

y se deduce

$$h = \frac{f(x_0)}{f'(x_0)}$$

si se entiende que la ecuación en la serie de Taylor no es una solución exacta y en buenas condiciones es una excelente aproximación, de esta manera:

$$x_0 + h = x_0 - \frac{f(x_0)}{f'(x_0)}$$

sería una mejor aproximación a x^* que la hecha inicialmente por x_0. Finalmente, la ecuación queda

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

el siguiente gráfico explicará cómo converge el método.

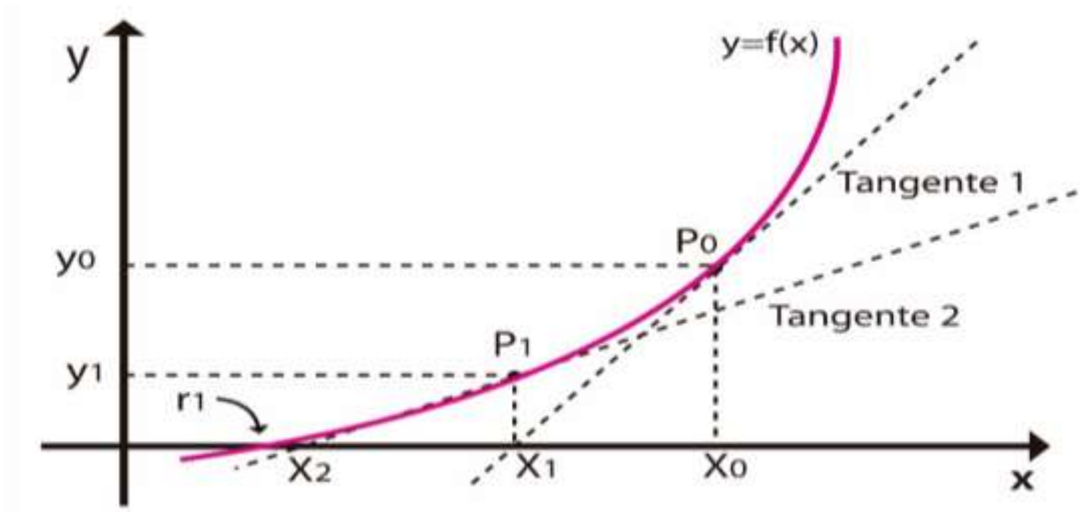

Figura 6. "Representación del método de Newton",

Elaboración propia

En las abscisas colocamos una aproximación (un valor que creamos cercano a la raíz) x_0, en este punto trazamos una recta tangente 1 a la gráfica, donde esta recta corta con la abscisa es el punto x_1,en este punto trazamos otra tangente 2, en donde corta con la abscisa en el punto x_2 y así sucesivamente hasta llegar lo más cercano a la raíz (r_1) o hasta donde permita el criterio de parada.

Para el segundo método el corrector, se usará a partir de la familia de iteraciones de Danby, suponiendo que el desarrollo de Taylor truncado en el término k-ésimo tiene una raíz δ_n,

$$0 = f(x_n) + f'(x_n)\delta_n + \ldots\ldots + \frac{f^k x_n}{k!}\delta_n^k$$

Esta familia tiene infinitos términos, recordemos que en el método de Danby usamos hasta k = 3, pero en este caso usaremos hasta k = 2 para reducir la cantidad de derivadas del método original y elaborar un método iterativo con mayores índices de eficiencia, el cual queda:

$$0 = f(x_n) + f'(x_n)\delta_n + \frac{1}{2} f''(x_n)\delta_n^2$$

Haciendo la resolución matemática hecha en páginas anteriores llegamos a estas ecuaciones

$$\delta_{n1} = -\frac{f(x_n)}{f'(x_n)} \tag{4.18}$$

$$\delta_{n2} = -\frac{f(x_n)}{f'(x_n) + \frac{1}{2} f''(x_n)\delta_{n1}} \tag{4.19}$$

y la ecuación iterativa es

$$x_{n+1} = x_n + \delta_{n2}$$

Usando a Newton como ecuación predictora (4.1) y las ecuaciones (4.18), (4.19) como correctora queda:

$$y_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$ método predictor

$$\delta_{n1} = -\frac{f(y_{n+1})}{f'(y_{n+1})}$$ método corrector

$$\delta_{n2} = -\frac{f(y_{n+1})}{f'(y_{n+1}) + \frac{1}{2} f''(y_{n+1})\delta_{n1}}$$

$$x_{n+1} = y_{n+1} + \delta_{n2}$$

5.1.1. Convergencia Local del método propuesto

Comenzaremos por hallar el orden de convergencia local del método predictor, luego del corrector, por último, utilizando el teorema de convergencia para métodos multipunto (ver teorema 4.3), se multiplican las dos convergencias encontrando la convergencia local del método propuesto. Entonces se obtendrá primero la convergencia local de Newton.

Teorema 4.5 Sea f: I⟶R una función de variable real definida en un entorno I de x^* una raíz simple de f(x) = 0, ($f(x^*) = 0$ y $f'(x^*) \neq 0$). Supongamos que f es dos veces derivable en I. Entonces el método de newton definido por $x_{n+1} = N_f(x_n)$

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

al menos, converge cuadráticamente.

Demostración: Al usar el teorema de Schröder (ver teorema 4.2), notemos que se necesita que f ” sea continua para satisfacer la hipótesis. El siguiente cálculo nos permite deducir lo siguente

$$N_f(x^*) = x^*, \quad N'_f(x^*) = 0, \quad N''_f(x^*) = \frac{f''(x^*)}{f'(x^*)} \neq 0,$$

por ende,

$$e_{n+1} = x_{n+1} - x^* = N_f(x_n) - x^* = \frac{N''_f(x^*)}{2!}(x_n - x^*)^2 + O_3,$$

de donde, usando la notación (4.6), se tiene

$$\lim_{n\to\infty} \frac{e_n + 1}{e_n^2} = \frac{N''_f(x^*)}{2!} = a_2,$$

donde a_2 es la constante del error, además se cumple la siguiente ecuación del error:

$$e_{n+1} = a_2 e_n^2 + O_3$$

con lo que se demuestra el orden de convergencia del método es cuadrático, un resultado muy conocido. Por consiguiente, se hallará la convergencia local del método corrector.

Teorema 4.6 Sea f: I⟶R una función de variable real definida en un entorno I de x^* una raíz simple de f(x) = 0, ($f(x^*) = 0$ y $f'(x^*) \neq 0$). Supongamos que f es tres veces derivable con continuidad en I. Entonces el método corrector por $x_{n+1} = Cr_f(x_n)$

$$\delta_{n1} = -\frac{f(x_n)}{f'(xn)}$$

$$\delta_{n2} = -\frac{f(x_n)}{f'(x_n) + \frac{1}{2} f''(x_n)\delta_{n1}}$$

$$x_{n+1} = x_n + \delta_{n2}$$

su convergencia es al menos, cúbica.

Demostración: Al usar el teorema de Schröder (ver teorema 4.2), el cálculo de las derivadas $D_{f,}$ nos permite deducir lo siguiente

$$Cr_f(x^*) = x^*, \quad Cr'_f(x^*) = Cr''_f(x^*) = 0, \quad Cr'''_f(x^*) = \frac{3f''(x)^2 - 2f'(x)\, f'''(x)}{2f'(x)^2} \neq 0,$$

por ende,

$$e_{n+1} = x_{n+1} - x^* = Cr_f(x_n) - x^* = \frac{Cr_f^{(''')}(x^*)}{3!}\,(x_n - x^*)^3 + O_4,$$

de donde, usando la siguiente notación (4.6), se deduce

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^3} = \frac{Cr_f^{(''')}(x^*)}{3!} = a_2^2 - a_3$$

Por lo tanto, la constante de error asintótico es $a_2^2 - a_3$, además se cumple la siguiente ecuación del error

$$e_{n+1} = (a_2^2 - a_3)\, e_n^3 + O_4$$

con lo que se demuestra el resultado tiene orden de convergencia local cúbica.

Al obtener el orden de convergencia del método predictor que es 2 y del método corrector que es 3 se realiza la multiplicación de las dos convergencias y el resultado será la convergencia del método propuesto, que en este caso es 6.

5.2. Resolución a un problema concreto

En este capítulo aplicaremos la parte conceptual de los métodos iterativos de Danby y el propuesto a un problema concreto, cuyo enunciado será de un libro muy usado por estudiantes, catedráticos y centros de estudios en la materia de Análisis Numérico.

Realizar esto es muy importante porque complementará lo teórico, se podrá entender numéricamente como se está realizando el problema por los dos métodos y observar sus diferencias.

5.2.1. Resolución por el método de Danby

El ejercicio es el 8.34 de la página 201 del libro de Steven C. Chapra y Raymond P. Canale (2011), *Métodos numéricos para Ingenieros*, séxta edición, Ciudad de México-México: edición en español por McGRAW-HILL/ INTERAMERICANA EDITORES, S.A DE C.V.

"Los sistemas mecánicos reales involucran la deflexión de los resortes no lineales. En la figura 7 se ilustra una masa m que se libera por una distancia h sobre un resorte no lineal. La fuerza de resistencia F del resorte está dada por la ecuación

$$F = -\left(k_1 d + k_2 d^{3/2}\right)$$

es posible usar la conservación de la energía para demostrar

$$0 = \frac{2k_2 d^{5/2}}{5} + \frac{1}{2}k_1 d^2 - mgd - mgh$$

despeje d, dado los siguientes valores de los parámetros: k_1 = 40000 g/s^2, k_2 = 40 g($s^2 m^{0.5}$), m = 95 g, gravedad = 9.81 m/s^2 y h = 0.43 m"

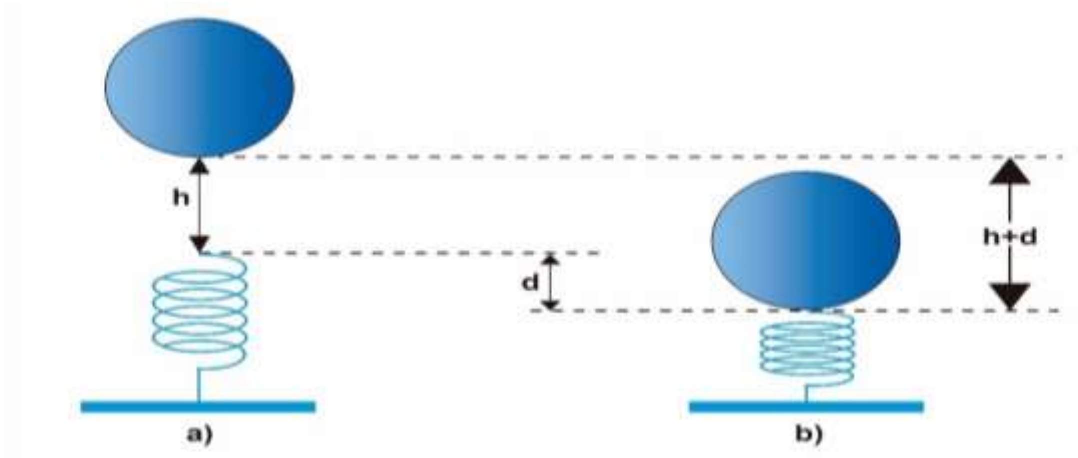

Figura 7. “Gráfica del problema”,

Fuente: (Chapra, S. C., Canale, R. P., 2011)

Lo primero que se realiza es la conversión al sistema SI (sistema internacional de unidades), de los datos que sean necesarios, en este caso: k_1 = 40000 g/s^2 = 40 Kg/s^2, k_2 = 0.04 kg ($s^2 m^{0.5}$), m = 95 g = 0.095 kg.

Luego se despejará d, para esto debemos resolver la ecuación no lineal de la forma f(x) = 0 dada por la relación al usar la conservación de la energía,

$$f(x) = \frac{2k_2 x^{5/2}}{5} + \frac{1}{2}k_1 x^2 - mgd - mgh$$

pero esta ecuación no tiene un modelo matemático para realizarla de forma analítica por lo que se requerirá un método iterativo.

En algunos casos las ecuaciones no lineales divergen de la solución y los polinomios de grado 2 en adelante pueden tener dos o más raíces, para salir de duda de esas dos complicaciones lo mejor es graficar la función al comenzar, sea de manera analítica (no tan recomendable) o con el uso de herramientas tecnológicas. La gráfica de esta función es la siguiente:

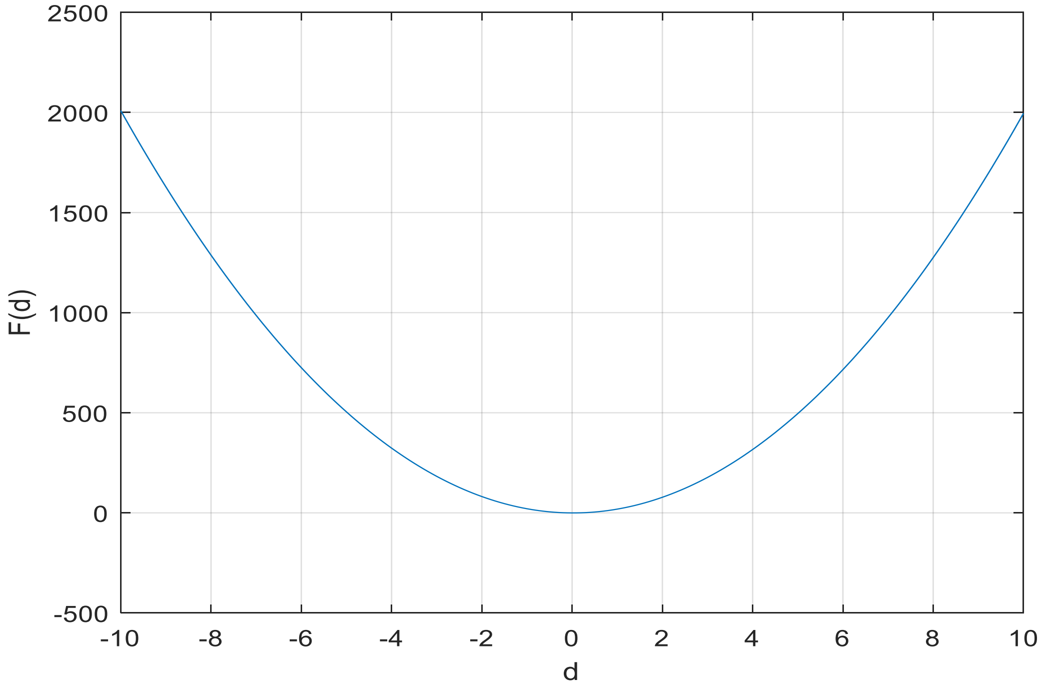

Figura 8. "Gráfica de la función del problema",

Elaboración propia

Donde podemos observar que hay dos raíces una negativa y una positiva, en este caso usaremos la raíz positiva, porque d en el problema, la usamos como distancia y siempre será positiva.

Resolveremos la ecuación no lineal por el método de Danby y sus fórmulas son:

$$L_{f(x)} = \frac{f(x)\, f''(x)}{f'(x)^2} \quad y \quad L_{f'(x)} = \frac{f'(x)\, f'''(x)}{f''(x)^2}$$

$$x_{n+1} = x_n - \frac{3}{2}\, \frac{(2 - L_f(x_n))^2}{6 - 9L_f(x_n) + 3L_f(x_n)^2 + L_f(x_n)^2\, L_{f'}(x_n)}\, \frac{f(x_n)}{f'(x_n)}$$

necesitaremos un criterio de parada, el cual será la tolerancia de 1×10^{-15} con 15 decimales y para la aproximación inicial observando la figura 8 se podría escoger un valor muy cercano a la solución pero para un mejor análisis se escogerá una aproximación inicial de 10.

Iteración 1

$x_0 = 10.000000000000000$

$$L_{f(x_0)} = \frac{f(x_0)\, f''(x_0)}{f'(x_0)^2}$$

$$L_{f(x_0)} = \frac{\left(\frac{2k_2\ x_0^{5/2}}{5} + \frac{1}{2}k_1x_0^2 - mgd - mgh\right) . \left(\frac{2k_2\,(\frac{5}{2}\cdot\frac{3}{2}\cdot x_0^{1/2})}{5} + k_1\right)}{\left(\frac{2k_2\,(\frac{5}{2}\cdot x_0^{\ 5/2})}{5} + k_1x_0 - mg\right)^2} =$$

$$L_{f(x_0)} = \frac{\left(1.995339405756269\, x\, 10^3\right) .\ (40.189736659610105)}{(4.003329610640674\, x\, 10^2)^2} = 0.500367671435913$$

$$L_{f'(x_0)} = \frac{f'(x)\, f'''(x)}{f''(x)^2}$$

$$L_{f'(x_0)} = \frac{\left(\frac{2k_2\, x_0^{\ 5/2}}{5} + \ k_1x_0 - mg\right) * \left(\frac{2k_2\,(\frac{5}{2}\cdot\frac{3}{2}\cdot\frac{1}{2}\ \cdot\ x_0^{-1/2})}{5}\right)}{\left(\frac{2k_2\,(\frac{5}{2}\cdot\frac{3}{2}\cdot x_0^{1/2})}{5} + k_1\right)^2} =$$

$$L_{f'(x_0)} = \frac{\left(4.003329610640674\, x\, 10^2\right) * (0.009486832980505)}{(40.189736659610105)^2} = 0.002351322948517$$

$$x_1 = x_0 - \frac{3}{2}\, \frac{(2 - L_f(x_0))^2}{6 - 9L_f(x_0) + 3L_f(x_0)^2 + L_f(x_0)^2\, L_{f'}(x_0)}\, \frac{f(x_0)}{f'(x_0)}$$

$$x_1 = 10.000000000000000 - \frac{3}{2} * \frac{(2 - 0.500367671435913)^2}{6 - 9 * 0.500367671435913 + 3 * 0.500367671435913^2 + 0.500367671435913^2 * 0.002351322948517} *$$

$$\frac{1.995339405756269\, x\, 10^3}{4.003329610640674\, x\, 10^2}$$

$$x_1 = 10.000000000000000 - \ 7.478008788196366$$

$$x_1 = 2.521991211803634$$

Iteración 2

$$L_{f(x_1)} = \frac{f(x_1)\, f''(x_1)}{f'(x_1)^2}$$

$$L_{f(x_1)} = \frac{\left(1.24619299210745\ x\ 10^2\right).\ (40.095284670133729)}{(1.001079032059433\ x\ 10^2)^2} = 0.498588062206441$$

$$L_{f'(x_1)} = \frac{f'(x_1)\, f'''(x_1)}{f''(x_1)^2}$$

$$L_{f'(x_1)} = \frac{\left(1.001079032059433\ x\ 10^2\ \right).\ (0.01889076179812)}{(40.095284670133729)^2} = 0.001176335579234$$

$$x_2 = x_1 - \frac{3}{2}\,\frac{(2 - L_f(x_1)\,)^2}{6 - 9L_f(x_1) + 3L_f(x_1)^2 + L_f(x_1)^2\ L_{f'}(x_1)}\,\frac{f(\,x_1)}{f'(\,x_1)}$$

$$x_2 = 2.521991211803634 - \frac{3}{2} * \frac{(2 - 0.498588062206441\,)^2}{6 - 9*0.498588062206441 + 3*0.498588062206441^2 + 0.498588062206441^2*0.001176335579234} * \frac{1.24619299210745\ x\ 10^2}{1.001079032059433\ x\ 10^2}$$

$$x_2 = 2.521991211803634 - 1.863527949279196$$

$$x_2 = 0.658463262524438$$

Iteración 3

$$L_{f(x_2)} = \frac{f(x_2)\, f''(x_2)}{f'(x_2)^2}$$

$$L_{f(x_2)} = \frac{(7.662713252599635) * (40.048687449566419)}{25.427953098900307^2} = 0.47462221061 7149$$

$$L_{f'(x_2)} = \frac{f'(x_2)\, f'''(x_2)}{f''(x_2)^2}$$

$$L_{f'(x_2)} = \frac{(25.427953098900307) *(0.036970513265813)}{40.048687449564419^2} = 5.861250832128554 \text{ x } 10^{-4}$$

$$x_3 = x_2 - \frac{3}{2}\frac{(2 - L_f(x_2))^2}{6 - 9L_f(x_2) + 3L_f(x_2)^2 + L_f(x_2)^2\, L_{f'}(x_2)}\frac{f(x_2)}{f'(x_2)}$$

$$x_3 = 0.658463262524438 - \frac{3}{2} *$$

$$\frac{(2 - 0.474622210617149)^2}{6 - 9 * 0.474622210617149 + 3 * 0.474622210617149^2 + 0.474622210617149^2 * 5.861250832128554 \text{ x } 10^{-4}} *$$

$$\frac{7.662713252599635}{25.427953098900307}$$

$$x_3 = 0.658463262524438 - 0.437444576369392$$

$$x_3 = 0.221018686155046$$

Iteración 4

$$L_{f(x_3)} = \frac{f(x_3)\, f''(x_3)}{f'(x_3)^2}$$

$$L_{f(x_3)} = \frac{(0.370635773135802) * (40.028207574694719)}{7.912953713600938^2} = 0.236938823084601$$

$$L_{f'(x_3)} = \frac{f'(x_3)\, f'''(x_3)}{f''(x_3)^2}$$

$$L_{f'(x_3)} = \frac{(7.912953713600938) *(0.063812646761757)}{40.028207574694719^2} = 3.151469418399397 \text{ x } 10^{-4}$$

$$x_4 = x_3 - \frac{3}{2}\frac{(2 - L_f(x_3))^2}{6 - 9L_f(x_3) + 3L_f(x_3)^2 + L_f(x_3)^2\, L_{f'}(x_3)}\frac{f(x_3)}{f'(x_3)}$$

$$x_4 = 0.221018686155046 - \frac{3}{2} *$$

$$\frac{(2 - 0.236938823084601\ )^2}{6 - 9 * 0.236938823084601\ + 3 * 0.236938823084601^2 + 0.236938823084601^2 * 3.151469418399397 \text{ x } 10^{-4}} *$$

$$\frac{0.370635773135802}{7.912953713600938}$$

$$x_4 = 0.221018686155046 - 0.054110909514885$$

$$x_4 = 0.166907776640161$$

Iteración 5

$$L_{f(x_4)} = \frac{f(x_4)\, f''(x_4)}{f'(x_4)^2}$$

$$L_{f(x_4)} = \frac{(0.001058016234483)*(40.024512608916730)}{5.747088628975721^2} = 0.001282102049452$$

$$L_{f'(x_4)} = \frac{f'(x_4)\, f'''(x_4)}{f''(x_4)^2}$$

$$L_{f'(x_4)} = \frac{\left(5.747088628975721\ \right)(0.073431596208898)}{40.024512608916730^2} = 2.634382054199095 \text{ x } 10^{-5}$$

$$x_5 = x_4 - \frac{3}{2}\, \frac{(2 - L_f(x_4)\ )^2}{6 - 9L_f(x_4) + 3L_f(x_4)^2 + L_f(x_4)^2\ L_{f'}(x_4)}\, \frac{f(\ x_4)}{f'(\ x_4)}$$

$$x_5 = 0.166907776640161 - \frac{3}{2} *$$

$$\frac{(2 -\ 0.001282102049452\ )^2}{6 - 9 * 0.001282102049452\ + 3 * 0.001282102049452^2 + 0.001282102049452^2 * 2.634382054199095 \text{ x } 10^{-5}} *$$

$$\frac{0.00105801623448}{5.747088628975721}$$

$$x_5 = 0.166907776640161 - 1.842142023271634 \text{ x } 10^{-4}$$

$x_5 = 0.166723562437834$

Iteración 6

$$L_{f(x_5)} = \frac{f(x_5)\, f''(x_5)}{f'(x_5)^2}$$

$$L_{f(x_5)} = \frac{(2.793321129956894 \text{ x } 10^{-13}\ ) * (40.024499078039312)}{5.739715546558111^2} = 3.393642048001203 \text{ x } 10^{-13}$$

$$L_{f'(x_5)} = \frac{f'(x_5)\, f'''(x_5)}{f''(x_5)^2}$$

$$L_{f'(x_5)} = \frac{(5.739715546558111^2) * (0.073472152589245)}{40.024499078039312^2} = 2.632457226587443 \text{ x } 10^{-4}$$

$$x_6 = x_5 - \frac{3}{2} \frac{(2 - L_f(x_5)\ )^2}{6 - 9L_f(x_5) + 3L_f(x_5)^2 + L_f(x_5)^2\ L_{f'}(x_5)} \frac{f(\ x_5)}{f'(\ x_5)}$$

$$x_6 = 0.166723562437834 - \frac{3}{2} *$$

$$\frac{(2 - 3.393642048001203 \text{ x } 10^{-13}\ )^2}{6 - 9 * 3.393642048001203 \text{ x } 10^{-13} + 3 * (3.393642048001203 \text{ x } 10^{-13})^2 + (3.393642048001203 \text{ x } 10^{-13})^2 * 2.632457226587443 \text{x } 10^{-4}}$$

$$* \frac{2.793321129956894 \text{ x } 10^{-13}}{5.739715546558111}$$

$x_6 = 0.166723562437834 - 4.866654292010022 \text{ x } 10^{-14}$

$x_6 = 0.16672356243778$

para concluir el ejercicio, d la igualamos con x_6 y calculamos la fuerza.

F= $-$ ($k_1\, x_6 + k_2\, {x_6}^{3/2}$)

F= $-$ (40 * 0.166723562437785 + 0.04 * $0.166723562437785^{3/2}$)

F= $-$ 6.671665546556169 N

quedando el signo negativo porque la fuerza es hacia abajo.

Obtenido el resultado de cada iteración hallaremos los errores absoluto y relativo porcentual que hubo en cada una, en la tabla que se mostrará a continuación:

Iteración (i)	x_i	x_{i+1}	$\lvert x_{i+1} - x_i \rvert$	$e_r = \frac{\lvert x_{i+1} - x_i \rvert}{x_{i+1}} * 100\%$
1	10.000000000000000	2.521991211803634	7.478008788196366	2.965120874806052 x 10^2
2	2.521991211803634	0.658463262524438	1.863527949279196	2.830116811885209 x 10^2
3	0.658463262524438	0.221018686155046	0.437444576369392	1.979219874931850 x 10^2
4	0.221018686155046	0.166907776640161	0.054110909514885	32.419645509713746
5	0.166907776640161	0.166723562437834	1.842142023269766 x 10^{-4}	0.110490802639647
6	0.166723562437834	0.166723562437785	4.901634653720066 x 10^{-14}	2.939977158626978 x 10^{-11}

Tabla 1. “Resultados de iteraciones y errores absoluto y porcentual para el método de Danby”, Elaboración propia

Para hallar el orden de convergencia se usó el método de ACOC y se colocarán sus resultados en la siguiente tabla, con x_0 = 10.000000000000000

Iteraciones (i)	x_i	$\frac{\ln(\lvert x_{i+1}-x_i\rvert / \lvert x_i - x_{i-1}\rvert)}{\ln(\lvert x_i - x_{i-1}\rvert / \lvert x_{i+1} - x_{i-2}\rvert)}$
1	2.521991211803634	_
2	0.658463262524438	_
3	0.221018686155046	1.043023964343151
4	0.166907776640161	1.442039461699249
5	0.166723562437834	2.719102960777168
6	0.166723562437785	3.881012765537152

Tabla 2. “Resultado del ACOC por cada iteración para el método de Danby”,

Elaboración propia

En la siguiente gráfica observaremos la tasa de convergencia del método.

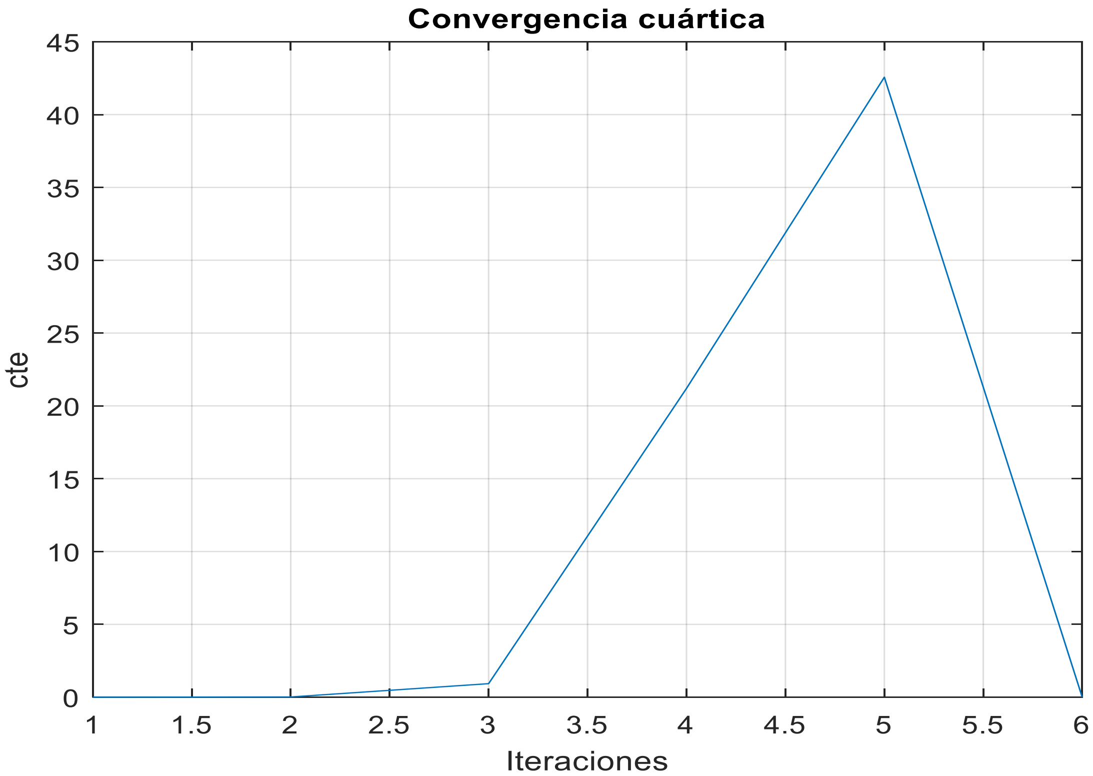

Figura 9. “Representación gráfica de la tasa de convergencia del método de Danby”, Elaboración propia

Los siguientes desarrollos y resultados serán de los índices de eficiencia y la orden de convergencia del método.

Eficiencia y convergencia del método

- Orden de convergencia:

  En ACOC = 3.881012765537152
- Índice de eficiencia:

  I= $p^{1/d}$ = $4^{1/6}$ = 1.259921049894873
- Índice de eficiencia computacional

  IC= $p^{1/(d+op)}$ = $4^{1/(6+14)}$ = $4^{1/20}$ = 1.071773462536293
- Optimalidad

  2^{d-1} = 2^{6-1} = 2^5 = 32 por lo tanto, el método es no óptimo.
- Método sin memoria
- Método con derivadas

5.2.2. Resolución por el método propuesto

Se realizará el mismo ejercicio, pero en esta ocasión por el método “Danby-Castillo” y con lo explicado anteriormente procederemos a la resolución:

Las ecuaciones son de la forma:

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

$$\delta_{n1} = -\frac{f(y_{n+1})}{f'(y_{n+1})}$$

$$\delta_{n2} = -\frac{f(y_{n+1})}{f'(y_{n+1}) + \frac{1}{2} f''(y_{n+1})\delta_{n1}}$$

$$x_{n+1} = y_{n+1} + \delta_{n2}$$

La función y los datos son los siguientes:

$f(x) = \frac{2k_2 x^{5/2}}{5} + \frac{1}{2} k_1 x^2 - mgd - mgh$; $k_1 = 40\ Kg/s^2$, $k_2 = 0.04\ kg\ (s^2 m^{0.5})$, $m = 95\ g = 0.095\ kg$, $h = 0.43\ m$ y gravedad $= 9.81\ m/s^2$

Aplicaremos el criterio de parada anterior, la misma cantidad de decimales y aproximación inicial.

Iteración 1

$$x_0 = 10.000000000000000$$

$$y_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$y_1 = 10 - \frac{1.995339405756269 \text{ x } 10^3}{4.003329610640674 \text{ x } 10^2}$$

$$y_1 = 5.015800346659578$$

$$\delta_{n1} = -\frac{f(y_1)}{f'(y_1)}$$

$$\delta_{n1} = -\frac{4.98991358794689 \text{ x } 10^2}{2.001493989746635 \text{ x } 10^2}$$

$$\delta_{n1} = -\, 2.493094465189253$$

$$\delta_{n2} = -\frac{f(y_1)}{f'(y_1) + \frac{1}{2} f''(y_1)\delta_{n1}}$$

$$\delta_{n2} = -\frac{4.98991358794689 \text{ x } 10^2}{2.001493989746635 \text{ x } 10^2 + \frac{1}{2} * 40.134375895338316 * -2.493094465189253}$$

$$\delta_{n2} = -\, 3.323949815227897$$

$$x_1 = y_1 + \delta_{n2}$$

$$x_1 = 5.015800346659578 + (-\, 3.323949815227897)$$

$$x_1 = 1.691850531431681$$

Iteración 2

$$y_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$y_2 = 1.691850531431681 - \frac{55.329275449626230}{66.830095638224748}$$

$$y_2 = 0.863940965820052$$

$$\delta_{n1} = -\frac{f(y_2)}{f'(y_2)}$$

$$\delta_{n1} = -\frac{13.733091748676445}{33.657809413532213}$$

$$\delta_{n1} = -\, 0.408020961196453$$

$$\delta_{n2} = -\frac{f(y_1)}{f'(y_1) + \frac{1}{2} f''(y_1)\delta_{n1}}$$

$$\delta_{n2} = -\frac{13.733091748676445}{33.657809413532213 + \frac{1}{2} * 40.055769054832872 * -0.408020961196453}$$

$$\delta_{n2} = -\, 0.538848187069217$$

$$x_2 = y_2 + \delta_{n2}$$

$$x_2 = 0.863940965820052 + (-\, 0.538848187069217)$$

$$x_2 = 0.325092778750836$$

Iteración 3

$$y_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

$$y_3 = 0.325092778750836 - \frac{1.410961716739723}{12.079175464044598}$$

$$y_3 = 0.208283339154594$$

$$\delta_{n1} = -\frac{f(y_3)}{f'(y_3)}$$

$$\delta_{n1} = -\frac{0.273107608410180}{7.403185825990489}$$

$$\delta_{n1} = -0.036890551558408$$

$$\delta_{n2} = -\frac{f(y_3)}{f'(y_3) + \frac{1}{2}f''(y_3)\delta_{n1}}$$

$$\delta_{n2} = -\frac{0.273107608410180}{7.403185825990489 + \frac{1}{2} * 40.027382841725370 * -0.036890551558408}$$

$$\delta_{n2} = -0.040977186391126$$

$$x_3 = y_3 + \delta_{n2}$$

$$x_3 = 0.208283339154594 + (-0.040977186391126)$$

$$x_3 = 0.167306152763468$$

Iteración 4

$$y_4 = x_3 - \frac{f(x_3)}{f'(x_3)}$$

$$y_4 = 0.167306152763468 - \frac{0,003350695139403}{5.763033444970739}$$

$$y_4 = 0.1667247410511841$$

$$\delta_{n1} = -\frac{f(y_4)}{f'(y_4)}$$

$$\delta_{n1} = -\frac{6.764937222292211 \text{ x } 10^{-6}}{5.739762719993427}$$

$$\delta_{n1} = -1.178609213012896 \text{ x } 10^{-6}$$

$$\delta_{n2} = -\frac{f(y_4)}{f'(y_4) + \frac{1}{2}f''(4)\delta_{n1}}$$

$$\delta_{n2} = -\frac{6.764937222292211 \text{ x } 10^{-6}}{5.739762719993427 + \frac{1}{2} * 40.024499164634463 * - \ 1.178609213012896 \text{ x } 10^{-6}}$$

$\delta_{n2} = -1.178614056335742 \text{ x } 10^{-6}$

$x_4 = y_4 + \delta_{n2}$

$x_4 = 0.1667247410511841 + 1.178614056335742 \text{ x } 10^{-6}$

$x_4 = 0.166723562437785$

Iteración 5

$$y_5 = x_4 - \frac{f(x_4)}{f'(x_4)}$$

$$y_5 = 0.166723562437785 - \frac{1.110223024625157 \text{ x } 10^{-6}}{5.739715546556164}$$

$y_5 = 0.16672356437785$

$$\delta_{n1} = -\frac{f(y_5)}{f'(y5)}$$

$$\delta_{n1} = -\frac{-5.551115123125783 \text{x } 10^{-17}}{5.739715546556164}$$

$\delta_{n1} = 9.6714115500101047 \text{x } 10^{-18}$

$$\delta_{n2} = -\frac{f(y_5)}{f'(y_5) + \frac{1}{2} f''(y_5)\delta_{n1}}$$

$$\delta_{n2} = -\frac{-5.551115123125783 \text{x } 10^{-17}}{5.739715546556164 + \frac{1}{2} * 40.024499078039305 * 9.6714115500101047 \text{x } 10^{-18}}$$

$\delta_{n2} = 9.67141155010147 \text{ x } 10^{-18}$

$x_5 = y_5 + \delta_{n2}$

$x_5 = \ 0.16672356437785 + 9.67141155010147 \text{ x } 10^{-18}$

$x_5 = 0.16672356437785$

de donde, d = x_5

la fuerza F= $-$ 6.671665546556169 N

Obtenido el resultado de cada iteración, se encontrará los errores tanto el absoluto como el relativo porcentual que hubo en cada una, en la tabla que se mostrará a continuación:

Iteración (i)	x_i	x_{i+1}	$\lvert x_{i+1} - x_i \rvert$	$e_r = \frac{\lvert x_{i+1} - x_i \rvert}{x_{i+1}} * 100\%$
1	10.000000000000000	1.691850531431681	8.308149468568320	4.910687625305635 x 10^2
2	1.691850531431681	0.325092778750836	1.366757752680845	4.204208281502255 x 10^2
3	0.325092778750836	0.167306152763468	0.157786625987368	94.310115546343127
4	0.167306152763468	0.166723562437785	5.825903256831688 x 10^{-4}	0.349434907198797
5	0.166723562437785	0.166723562437785	2.775557561562891 x 10^{-17}	1.664766227988096 x 10^{-14}

Tabla 3. "Resultados de iteraciones y errores absoluto y porcentual para el método propuesto", Elaboración propia

Para encontrar el orden de convergencia se usó el método de ACOC y se mostrarán sus resultados en la siguiente tabla, con x_0 = 10.000000000000000

Iteraciones (i)	x_i	$\frac{\ln(\lvert x_{i+1} - x_i \rvert / \lvert x_i - x_{i-1} \rvert)}{\ln(\lvert x_i - x_{i-1} \rvert / \lvert x_{i+1} - x_{i-2} \rvert)}$
1	1.691850531431681	_
2	0.325092778750836	_
3	0.167306152763468	1.196231306919591
4	0.166723562437785	2.594551525812087
5	0.166723562437785	5.476209614544238

Tabla 4. "Resultado del ACOC por cada iteración para el método propuesto"

Elaboración propia

la tasa de convergencia del método se mostrará en la siguiente gráfica.

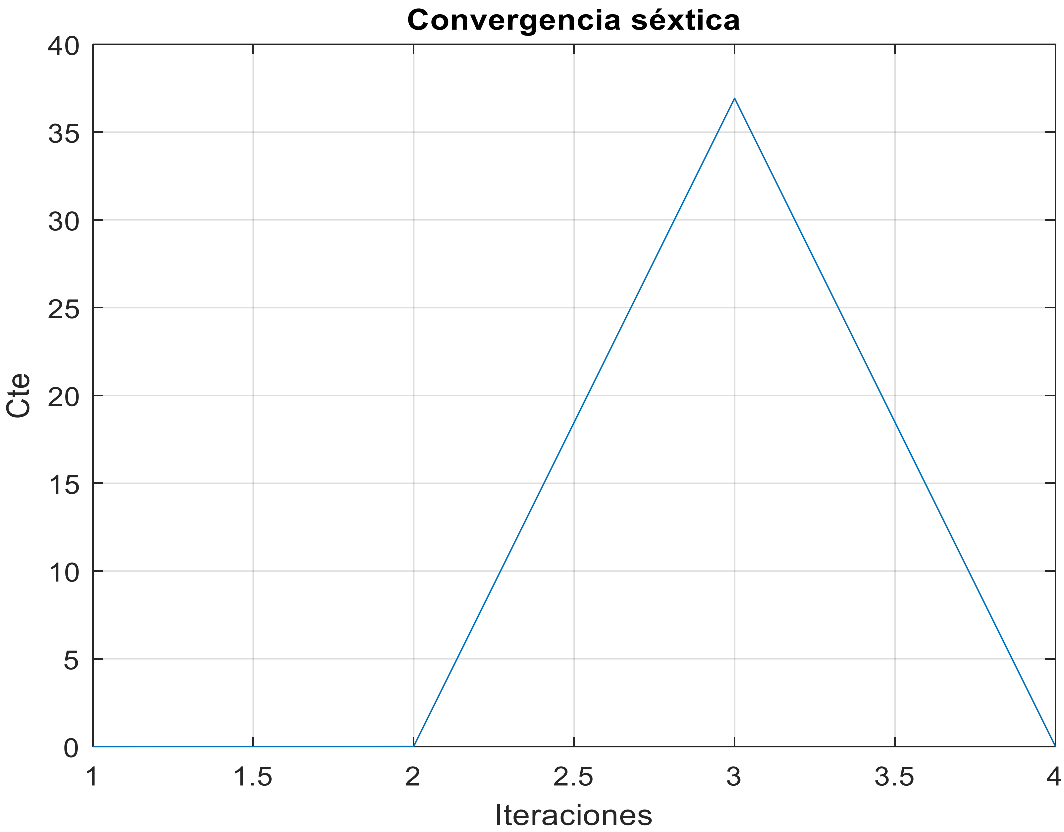

Figura 10. “Representación gráfica de la tasa de convergencia del método propuesto”

Elaboración propia

El desarrollo de resultados del índice de eficiencia y el orden de convergencia del método a continuación:

Eficiencia y convergencia del método

- Orden de convergencia:

  En ACOC = 5.476209614544238
- Índice de eficiencia:

  I= $p^{1/d}$ = $4^{1/5}$ = 1.319507910772894
- Índice de eficiencia computacional

  IC= $p^{1/(d+op)}$ = $4^{1/(5+6)}$ = $4^{1/11}$ = 1.134312522195463
- Optimalidad

$2^{d-1} = 2^{5-1} = 2^4 = 16$ por lo tanto el método es no óptimo

- Método sin memoria
- Método con derivada

5.3. Elaboración del algoritmo del método propuesto

En esta sección se explicará como se realizó el algoritmo del programa en el lenguaje de MATLAB.

Habrá que colocar las variables de salida y entrada del método. Las variables de salida se colocan dentro del corchete que son la solución (so), el número de iteraciones (iter) que se realizarán y la diferencia de la respuesta de la iteración actual con la anterior (incre). Del otro lado del igual se coloca el nombre del archivo y dentro del paréntesis las variables de entrada que son la función (funcion), una aproximación inicial (x_0), el valor de la tolerancia (tol) y el número máximo de iteraciones (maxiter).

Luego inicializamos el criterio de parada, incre = tol + 1 para empezar con un valor mayor a la tolerancia e iter = 1, es decir, que comenzará de la iteración 1. Después usamos un bucle iterativo while, el cual ordenará que, si se cumplen ciertas condiciones el programa seguirá, en este caso incre > tol e iter < maxiter y si no se cumple esta condición el programa parará.

Por consiguiente, programamos el método predictor, pero primero se almacena la función y su derivada en un vector [f, df] = feval(funcion, x_0), de donde la ecuación de Newton en el programa queda y = $x_0 -$ (f / df). De donde, feval es una función de Matlab R2018a que evalúa la aproximación x_0 en la función que es f y en la derivada de la función que es df. Luego se programa el método corrector que al igual que el predictor, primero almaceno las funciones, en este caso una más, porque también está la segunda derivada con el valor de y evaluado [fy, dfy, sdfy] = feval(funcion, y). En este caso y es la respuesta de la primera iteración hallada por el método de Newton y uso feval para evaluar y en la función fy en la derivada de la función dfy y en la segunda derivada de la función sdfy.

Después se programan las dos ecuaciones sn_1 y sn_2 quedando $sn_1 = -$ fy / dfy y $sn_2 = -$ fy / (dfy + (0.5 * sdfy * sn_1)), en las cuales usaremos los datos almacenados anteriormente. Por consiguiente, se programa la ecuación x = y + sn_2, se finaliza el criterio de parada incre = abs(x - x_0), se deja inicializado un valor I(iter) = incre para usarlo después para el ACOC y también se actualiza el programa con iter = iter +1 y x_0 = x.

Luego, se programa el ACOC que usa el siguiente código en el programa ACOC = log(I(3:end) ./I(2:end −1)) ./log(I(2:end −1) ./I(1:end − 2)), del cual la función log actúa como logaritmo natural en el programa, con el cual hallamos el orden de convergencia del método pero empezará a dar resultados a partir de las primeras 4 iteraciones consecutivas x_{k+1}, x_k, x_{k-1} y x_{k-2}.

Al final, usamos la sentencia condicional if para comunicar al programa, si el incre > tol, es decir, al no cumplir el criterio de parada muestre que se necesita más iteraciones con el siguiente código:

```
if  incre > tol
     disp ('necesito más iteraciones')
```

Completamos el condicional if sino se cumple, con else que significa caso contrario y mostramos la orden que en este caso sería la solución, luego cerramos con end todas las sentencias como muestra el siguiente código:

```
else
       sol = x;
end
end
```

Por último, se hace la función en otro archivo m, en donde las variables de salida son la función, su primera y segunda derivada. Al otro lado del igual se coloca el nombre del archivo con la variable de entrada dentro de un paréntesis. Luego escribimos los códigos de las variables de salida en el programa, finalizando con un end.

6. Conclusiones

El principal motivo de la realización de este TFM, fue plasmar en su desarrollo, nuevos y muy útiles conocimientos adquiridos, más la investigación que se debe realizar, la cual complementará y será necesaria colaborando a un mejor análisis. El objetivo principal era crear un nuevo método iterativo, el que se lo pudo hacer estudiando un desarrollo matemático relativamente reciente como el de Danby.

Para la realización de este método que se propone, se hicieron búsquedas de artículos relacionados a la propuesta de este trabajo para observar si alguno de estos coincidía o era similar, también se hizo el estudio teórico, el cual será esencial para un mejor entendimiento del ejemplo práctico que se desarrollaría para los dos métodos, en el cual los conceptos teóricos ayudarán a analizar los resultados numéricos y gráficas.

Para luego, con los resultados obtenidos poder analizarlos teniendo información para saber si se cumplieron los objetivos y a partir de esto, realizar las conclusiones que se darán a continuación:

- En el desarrollo matemático para obtener el método iterativo de Danby, se usa k = 3 en el k-ésimo término de su familia de métodos iterativos (4.14), junto con las notaciones de grado de convexidad logarítmicas (4.15), (4.16), creando un método, el cual su máxima cantidad de derivadas es 3, lo cual es una cantidad muy elevada y limita mucho la cantidad de problemas que el método puede resolver, por eso Danby lo desarrolló especialmente para la ecuación de Kepler (4.13), la cual no tiene problemas con esta cantidad de derivadas. En cambio, el método iterativo propuesto usa como predictor el método de newton y como corrector un método iterativo de

Danby, pero con k = 2, el cual tiene como máximo dos derivadas, disminuyendo la cantidad de derivadas y no solo limitándolo para resolver una o un grupo reducido de ecuaciones f(x) = 0.

- Al realizar el problema por ambos métodos pudimos percatarnos que al hacerlo por el método de Danby analíticamente era muy extenso, es decir muchas operaciones matemáticas y evaluaciones funcionales, dándonos una pauta que, por ende, sus índices de eficiencia serían bajos y con un alto coste de computo. El método propuesto era menos complicado de resolver analíticamente, por consiguiente, sus índices de eficiencia fueron superiores y su coste de ordenador más bajo.

- El orden de convergencia de Danby, que es un método iterativo de un punto, fue hallado de manera analítica por el teorema de Schröder (ver teorema 4.2) y el método ACOC (4.11), donde el resultado por ambas tiende a tener orden de convergencia 4. En el método propuesto que es un método iterativo multipunto, se calculó el orden de convergencia de forma analítica usando el teorema de Schröder en combinación con el teorema para encontrar la convergencia a métodos multipuntos (ver teorema 4.3) y también se usó otro método el ACOC y en ambas resoluciones tiende a tener orden de convergencia 6. Al resolver el problema por el método de Danby se necesitaron 6 iteraciones y por el método propuesto 5, con el mismo criterio de parada, por lo que el método propuesto al tener menor cantidad de iteraciones para alcanzar el valor numérico deseado corrobora lo que se demostró en los teoremas, que el método propuesto tiene mayor orden de convergencia.

- Antes de resolver una ecuación no lineal por cualquier método iterativo, lo recomendable es realizar la gráfica de la función para escoger una estimación inicial adecuada porque la función a resolver puede complicar de varias formas como: raíces con multiplicidad ≥ 2, también haber más de una raíz, que la estimación inicial cause una división por 0, entre otras causas.

Al analizar los resultados de los métodos de Danby y el propuesto se realizaron las conclusiones y a partir de estas se harán los subcapítulos de ventajas, desventajas y líneas futuras para el método propuesto.

6.1 Ventajas

- El método propuesto tiene su orden de convergencia 6 y solo tiene una cantidad máxima de derivadas que son 2, esta es la ventaja de que el método pertenezca a la clasificación de métodos multipuntos porque en los métodos de un punto no se podría tener ese número de convergencia con esa cantidad de derivadas.

- Para problemas de ecuaciones no lineales, los cuales pueden ser muy complicados o tener la aproximación inicial muy alejada de la raíz, pero si tienen condiciones de convergencia, el método propuesto sería excelente para obtener con una alta velocidad de convergencia la solución.

- La elaboración del algoritmo del método propuesto en Matlab R2018a fue de mucha utilidad tanto para el cálculo de los resultados, como para las simulaciones que se realizaron. También se realizaron en el software diferentes gráficas que ayudaron a comprender los datos numéricos.

6.2 Desventajas

- El método propuesto tiene derivadas, en este caso son dos, la dificultad es cuando su derivada sean funciones que sean muy complicadas u otras funciones que al evaluar el valor inicial en la derivada nos dé una división por cero, la que, en este caso no tendrá solución.

- Para ecuaciones no lineales con 2 raíces o más, el método propuesto dependiendo de la estimación inicial, dará de respuesta a la raíz más cercana y si resuelve algún problema que tenga la raíz de multiplicidad igual o mayor a 2, su convergencia es de orden lineal.

- El método no está diseñado para sistemas de ecuaciones.

6.3 Líneas futuras

Del método Danby-Castillo hemos escrito sus ventajas y desventajas en esta parte se colocará la propuesta de cómo mejorar el método y proponer una nueva versión de este:

- En lo que respecta a derivadas como inconveniente, si se las quiere evitar se podría usar diferencias divididas, en este caso hay bastantes maneras de derivar numéricamente, con lo que se puede variar matemáticamente el método y hacerlo sin derivadas, pero su convergencia e índices de eficiencia cambiarán.

- Hacer un estudio de dinámica compleja para garantizar estimaciones y el método no diverja.

- Para mejorar el orden de convergencia del método propuesto se puede en vez de usar como predictor el método de Newton (que es óptimo), utilizar otro de mayor orden de convergencia, pero como consecuencia usando más derivadas, más evaluaciones funcionales y perdiendo optimalidad.

7. Bibliografía

- A. Corder and J. R. Torregrosa. (2007). *Variants of Newton´s method using fifth-order quadrature formulas.* Applied Mathematics and Computation 190, no: 1 686-698.
- Abu, I., & Alshaikh. (2005). A New Iterative Method for Solving Nonlinear Equations. *Proceedings of World Academy of Science, Engeneering and Technology, 5*.
- Ahmad, N., Rafiq, N., & Akram, S. (2009). An Efficient Three-Step Iterative Method for Non-Linear Equations. *International Journal of Mathematical Analysis, 3*(40), 1989 - 1996.
- Alan Hirshfeld . (2017). *Bass Bulletin of the Aas.* BASS CITATION: BAAS 49, 017.
- Biazar, J., & Ghanbari, B. (2008). A New Computational Approach for Nonlinear Equations. *International Mathematical Forum, 3*(20), 955-960.
- Burden, R., & Faires, J. (1998). *Análisis Numérico,* . Madrid, España: International Thompson Editores.
- Cabezas, J. A. (2018). *Métodos Numericos.* Grupo Anaya, S. A.
- Calderón, G. (2007). Métodos iterativos para resolver ecuaciones no lineales. *Revista Notas de Matemática, 3*(250), 33-44.
- Casalderrey, M. (2000). *Cardano y Tartaglia. Las matemáticas en el renacimiento italiano.* Nivola: Madrid. Obtenido de https://revistasuma.es/IMG/pdf/56/089-096.pdf
- Chapra, S. C., Canale, R. P. (2011). *Métodos númericos para ingenieros.* Cuidad de Mexico: McGRAW-HILL/INTERAMERICANA EDITORES. S.A DE C.V .

- Corbalán, F. (2010). *Galois: revolución y matemáticas* (Segunda edición ed.). Logroño, España: NIVOLA ISBN: 9788492493562.
- D. Kincaid. (1991). *Análisis Numérico.* España: Addison- Wesley Iberoamericana. Obtenido de https://evflores.files.wordpress.com/2014/02/analisis-numerico-richard-l-burden-7ma.pdf
- Dalcín, M., & Olave, M. (2007). *Ecuaciones de Segundo Grado: Su historia.* Uruguay: Acta Latinoamericana de Matemática. Obtenido de http://funes.uniandes.edu.co/5169/1/Dalc%C3%ADnEcuacionesALME2007.pdf
- Danby, J. M. (1988). Fundamentals of Celestial Mechanics, 2nd. edition. *BAAS BULLETIN OF THE AAS*, 49, 017. Obtenido de https://aasjournals.github.io/aas-obits-mirror/j-m-anthony-danby-1929-2009.html
- Ezquerro, J. A., Gutiérrez, J. M., Hernández, M. A., & Salanova, A. (2001). El método de Halley: Posiblemente, el método mas redescubierto del mundo. *Servicio de Publicaciones, Universidad de la Rioja*.
- H.T. Kung and J.F. Traub . (s.f.). *Optimal order of one-point and multipoint iteration.* Applied Mathematics and computation 21, 643-651.
- In Yun, B. (2011). A quadratically convergent iterative method for nonlinear equations. *J. Korean Math. Soc., 48*(3), 487-497. doi:10.4134/JKMS.2011.48.3.487
- J. M. Danby, T. M. (1983). The solution of Kepler's equation I. *Celestial Mechanics, 31*, 95-107.
- J.F Traub. (1964). *Iteractive methods for the solution of equations,* . Englewood cliffs. N.J Prentice- Hall .
- Kincaid, D. (1991). *Análisis Numérico.* España: Addison We.
- Mamta, V. K. (2005). On some third order iterative methods for solving non-linear equations. *Appl. Math Comput, 171*, 272-280.

- Mario Dalcín, m. O. (2011). *Ecuaciones de segundo grado: Su historia.* Acta Latinoamericana de Matemática Educativa Vol. 20. Obtenido de http://funes.uniandes.edu.co/5169/1/Dalc%C3%ADnEcuacionesALME2007.pdf
- Martin casaderrey, F. (2000). *Tartaglia: El desafío de una ecuación.* Suma 56.
- Muñoz, J. M. (2011). *Historia de matemáticas Abel y la imposibilidad de resolver la quíntica por radicales.* Madrid, España: Pensamiento matemático. Obtenido de http://www2.caminos.upm.es/Departamentos/matematicas/revistapm/revista_impresa/numero_1/abel_y_la_quintica.pdf
- Nieves , A., & Dominguez, F. (2014). *2014) Metodos numéricos aplicados a la ingeniería,.* México: Grupo Editorial Patria, S.A de C.V. Obtenido de https://editorialpatria.com.mx/pdffiles/9786074383171.pdf
- Ostrowki, A. M. (1964). *Solution of equations and system of equations.* Pretice-Hall.
- S. Weerakoon and T. G. I. Fernando A. (2000). *Variant of Newton´s method with accelerated third- order convengence.* Applied mathematics letters 13. no. 8, 87-93.
- Sánchez , J. (2011). *Historia de Matemáticas Abel y la imposibilidad de resolver la "quintica" por radicales.*
- Stewart , I. N. (2008). *Historia de las matemáticas en los últimos 10.000 años ,.* Barcelona, España: Crítica. Obtenido de http://www.librosmaravillosos.com/historiadelasmatematicasenlosultimos10000anos/pdf/Historia%20de%20las%20matematicas%20-%20Ian%20Stewart.pdf
- Stewart, I. n. (2008). *Historia de las matematicas en los ultimos 10.000 años.* Barcelona, España: Crítica. Obtenido de

http://www.librosmaravillosos.com/historiadelasmatematicasenlosultimos10000anos/pdf/Historia%20de%20las%20matematicas%20-%20Ian%20Stewart.pdf

- Traub, H.T. Kung. J.F. (21 de 03 de 1974). *Optimal order of ane- point and ,multi-point iteration. Applied mathematics and computation.* KT 643-651. Obtenido de Applied mathematics and computation.

8. Anexos

8.1 Programa de resolución para el método de Danby

Fichero Danby.m

El siguiente código resuelve el método clásico y ayuda a simular sus datos.

```
function   [sol,iter,incre]=Danby(funcion,xo,tol,maxiter)
incre=tol+1; % Inicializamos con un valor el criterio de parada
iter=1; % Inicializamos el numero de iteraciones

while incre>tol && iter<maxiter % Condiciones del bucle
 [f,df,ddf,dddf]=feval(funcion,xo);
 lf=(f.*ddf)/df.^2;        % La ecuación de la notación
 lfd=(df.*dddf)/ddf.^2;    % La derivada de la notación
 g=(2-lf)^2/(6-9*lf+3*lf^2+lf^2*lfd);
 x=xo-((3/2)*g)*(f/df);    % La ecuación de Danby
 incre=abs(x-xo)+abs(feval(funcion,x));
 I(iter)=incre;
 iter=iter+1;
 xo=x;
end
 ACOC=log(I(3:end)./I(2:end-1))./log(I(2:end-1)./I(1:end-2));

if incre>tol
    disp('necesito mas iteraciones')
else
    sol=x;
end
```

8.2 Programa de resolución para el método propuesto

Fichero DanbyCastillo.m

El siguiente código resuelve el método propuesto y ayuda a simular sus datos.

```
function    [sol,iter,incre]=DanbyCastillo(funcion,xo,tol,maxiter)
incre=tol+1; % Inicializamos con un valor el criterio de parada
iter=1;      % Inicializamos el numero de iteraciones

while incre>tol && iter<maxiter  % Condiciones del bucle
 [f,df]=feval(funcion,xo);
 y=xo-(f/df);                         % Método de newton (predictor)
 [fy,dfy,sdfy]=feval(funcion,y);
 sn1=-fy/dfy;                         % Método de Danby (corrector)
 sn2=-fy/(dfy+(0.5*sdfy*sn1));
 x=y+sn2;
 incre=abs(x-xo);
 I(iter)=incre;
 iter=iter+1;
 xo=x;
end
%Linea del programa para hallar orden de convergencia
ACOC=log(I(3:end)./I(2:end-1))./log(I(2:end-1)./I(1:end-2));

if incre>tol
    disp('necesito mas iteraciones')
else
    sol=x;
end
end
```

Estos códigos implementarán la función del método iterativo de Danby y propuesto

Función del método iterativo de Danby:

```
function [f,df,ddf,dddf]=eg3(x)
f=(2*0.2*0.04*x.^2.5)+(0.5*40*x.^2)-(0.095*9.81*x)-(0.095*9.81*0.43);
df=(2*0.2*0.04*2.5*x.^1.5)+(40*x)-(0.095*9.81);
ddf=(2*0.2*0.04*2.5*1.5*x.^0.5)+40;
dddf=(2*0.2*0.04*2.5*1.5*0.5*x.^-0.5);
end
```

Función del método iterativo propuesto:

```
function [f,df,sdf]=eg13(x)
f=(2*0.2*0.04*x.^2.5)+(0.5*40*x.^2)-(0.095*9.81*x)-(0.095*9.81*0.43);
df=(2*0.2*0.04*2.5*x.^1.5)+(40*x)-(0.095*9.81);
sdf=(2*0.2*0.04*2.5*1.5*x.^0.5)+40;
end
```

8.3 Comparación de resultados

Método iterativo	**Orden de convergencia**	**Número de iteraciones**	**Índice de eficiencia**	**Índice de eficiencia computacional**
Danby	4	6	1.259921049894873	1.071773462536293
Danby-Castillo	6	5	1.319507910772894	1.134312522195463