

Ecuaciones no lineales en física y su resolución mediante el uso de métodos iterativos multipaso de orden alto

Nonlinear equations in Physics, and their resolution using high order multi-step iterative methods

Santiago Quinga¹

¹Departamento de Ciencias Exactas, Universidad de las Fuerzas Armadas ESPE, S/N y Ambato, Av. Gral. Rumiñahui, Sangolquí, Ecuador.

E-mail: sant.david@hotmail.com

Recibido el 30 de Abril de 2021 | Aceptado el 6 de Agosto de 2021

Resumen

El presente trabajo proporciona al docente de física fundamentos teóricos y computacionales para resolver ecuaciones no lineales, muy comunes en la solución de problemas físicos. En el presente trabajo de investigación se resuelven tres problemas de física, los cuales son: una esfera flotando en agua, caída no libre de un paracaidista, compresión de un resorte real; haciendo uso de principios referentes a fluidos, cinemática y dinámica. Se obtienen ecuaciones no lineales difíciles y en algunos casos imposibles de ser resueltas mediante métodos analíticos. Para encontrar una solución aproximada a dichas ecuaciones se hace uso de métodos iterativos partiendo desde los métodos tradicionales como son Newton, Secante, Steffensen hasta la introducción de métodos multipaso con alto orden de convergencia como son Traub, Ostrowski y métodos de orden ocho diseñados a partir del método de Ostrowski. Finalmente, se realiza un análisis de los resultados obtenidos al aplicar todos estos métodos a cada uno de los problemas físicos seleccionados y de esta forma establecer qué método iterativo es más adecuado ante cada situación.

Palabras clave: Ecuaciones no lineales; Métodos iterativos multipaso; Newton; Métodos de Ostrowski; Física.

Abstract

This work provides the physics teacher with theoretical and computational foundations to solve nonlinear equations, very common in solving physical problems. In the present research three physics problems are solved, which are: a sphere floating in water, non-free fall of a parachutist, compression of a real spring; making use of principles related to fluids, kinematics and dynamics. Nonlinear equations are obtained which are difficult and, in some cases, impossible to be solved by means of analytical methods. To find an approximate solution to these equations we use iterative methods starting from traditional methods such as Newton, Secant, Steffensen to the introduction of multi-step methods with high order of convergence such as Traub, Ostrowski and methods of order eight designed from Ostrowski's method. Finally, an analysis of the results obtained by applying all these methods to each of the selected physical problems is carried out and, in this way, establish which iterative method is more appropriate in each situation.

Keywords: Nonlinear equations; Iterative multi-step methods; Newton; Ostrowski's method; Physics.

I. INTRODUCCIÓN

El presente trabajo proporciona al docente de física fundamentos teóricos y computacionales para resolver ecuaciones no lineales, muy comunes en la solución de problemas físicos; haciendo uso de principios referentes a fluidos, cinemática y dinámica se han planteado tres problemas cuyo modelamiento matemático lleva a la obtención de una ecuación no lineal, dichos ejemplos son: una esfera flotando en agua, caída no libre de un paracaidista, compresión de un resorte real. Al momento de encontrar la solución de una ecuación no lineal, el método que, por excelencia es utilizado por el docente, es el de Newton, sin embargo, este método no siempre es el más óptimo ante cada situación, hecho que se va a demostrar mediante un análisis de diferentes métodos iterativos. En forma general, una ecuación no lineal se puede escribir de la forma $f(x)=0$, y el objetivo será encontrar una solución de la misma, donde $f: I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ es una función real que está definida en un intervalo I abierto.

El desarrollo de los métodos numéricos ha permitido la aparición de técnicas computacionales que permiten resolver varios tipos de modelos matemáticos que no tienen una solución exacta, tal es el caso de los métodos iterativos los cuales nos van a permitir obtener la solución aproximada de una ecuación no lineal.

II. MARCO TEORICO

A. Métodos iterativos

Como mencionan Mathews & Fink (2000), una iteración consiste en realizar un proceso por repetidas ocasiones hasta que se obtiene un resultado con el menor error posible. En este sentido los métodos iterativos se utilizan para hallar raíces de ecuaciones y sistemas de ecuaciones sean estos lineales o no lineales, y en la solución numérica de ecuaciones diferenciales.

Para resolver una ecuación no lineal mediante el uso de métodos iterativos se va a realizar un proceso en el cual mediante una fórmula (función de punto fijo) se va a sustituir una primera aproximación de la raíz, con ello se obtendrá una nueva estimación de la raíz, este proceso se realizará varias veces obteniéndose una sucesión de valores (iterados) que se aproximarán cada vez más a la raíz de la ecuación estudiada (García, 2017).

García (2017) indica que un método iterativo de punto fijo se representa de la siguiente forma:

$$x_{n+1} = G(x_n); \quad n = 0, 1, 2, \dots, \quad (1)$$

donde G es la función de punto fijo y α al ser la raíz de la ecuación debe cumplir que $\alpha = G(\alpha)$ convirtiéndose de esta forma en un punto fijo de la función G ya que los puntos fijos son aquellos valores de x que satisfacen la ecuación $G(x) = x$.

B. Clasificación de métodos iterativos

Según Chicharro (2017), los métodos iterativos podrían clasificarse como métodos con memoria o sin memoria en función de los datos necesarios para obtener el valor del siguiente iterado; un método sin memoria solo depende del valor del último iterado:

$$x_{n+1} = \phi(x_n); \quad n = 0, 1, 2, \dots, \quad (2)$$

mientras que un método con memoria depende de varios iterados anteriores:

$$x_{n+1} = \phi(x_n, x_{n-1}, x_{n-2}, \dots); \quad n = 0, 1, 2, \dots, \quad (3)$$

Amorós (2020) menciona que los métodos iterativos se pueden clasificar según el número de pasos en cada iteración, obteniéndose de este modo métodos de un paso o multipaso; un método de un paso tiene la forma de la ecuación (1), mientras que un método multipaso viene descrito por:

$$\begin{aligned} y_n &= \psi(x_n) \\ x_{n+1} &= \phi(x_n, y_n); \quad n = 0, 1, 2, \dots \end{aligned} \quad (4)$$

El ultimo criterio que se puede emplear para clasificar un método iterativo es dependiendo de la presencia o ausencia de derivadas, obteniéndose métodos con derivadas o libres de derivadas (Chicharro, 2017).

C. Orden de convergencia

Sea $\{x_k\}_{k \geq 0}$ una sucesión de números reales tales que $\lim_{k \rightarrow \infty} x_k = \xi$. La convergencia se llama:

- Lineal, si existe M , $0 < M < 1$ y $k_0 \in \mathbb{N}$ tales que $|x_{k+1} - \xi| \leq M|x_k - \xi|$, para todo $k \geq k_0$;
- De orden p , si existe $M > 0$ y $k_0 \in \mathbb{N}$ tales que $|x_{k+1} - \xi| \leq M|x_k - \xi|^p$, para todo $k \geq k_0$.

Denotamos por $e_k = x_k - \xi$ el error de la k -ésima iteración. La ecuación $e_{k+1} = Me_k^p + O(e_k^{p+1})$, se llama ecuación del error, donde p es el orden de la convergencia (Artidiello, 2014).

Del orden de convergencia de un método iterativo depende el número de iteraciones que se debe realizar para obtener una solución aproximada de la ecuación no lineal, siendo los métodos de orden de convergencia alto quienes se aproximan más rápido a la solución, necesitando de este modo menos iteraciones (Amorós, 2020).

Para realizar una comparación entre métodos iterativos se puede utilizar parámetros relacionados al orden de convergencia tales como el índice de eficiencia:

$$I = p^{1/d} \quad (5)$$

donde p es el orden de convergencia del método y d es el número de evaluaciones de la función f y sus derivadas, realizadas en cada iteración (Ostrowski, 1966).

Por otro lado, Kung y Traub (1974) establecen una conjetura que permite saber si un método es óptimo o no, dicha conjetura establece que el orden de convergencia p de un método sin memoria con d evaluaciones funcionales por iteración cumple con:

$$p \leq 2^{d-1} \quad (6)$$

siendo un método óptimo aquel que cumple la igualdad en la expresión anterior.

D. Método de Newton

Como indica Ypma (1995) el método de Newton es uno de los más conocidos y utilizados al momento de resolver una ecuación no lineal, en donde la función de punto fijo tiene la siguiente forma:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}; \quad n = 0, 1, 2, \dots, \quad (7)$$

en ocasiones también es nombrado como método de Newton-Raphson.

Además, este método tiene una representación geométrica en donde, en cada etapa n , el valor x_{n+1} corresponde a la abscisa del punto de corte con el eje OX de la recta tangente a la curva $y = f(x)$ en el punto $(x_n; f(x_n))$, debido a ello este método también es conocido como el método de la tangente (Artidiello, 2014).

Como menciona Amorós (2020) en base a la teoría planteada anteriormente, el método de Newton tiene las siguientes características:

- Es un método sin memoria.
- Es un método de un paso.
- Es un método con derivada
- Tiene un orden de convergencia cuadrática
- Según la conjetura de Kung y Traub es un método óptimo

E. Método de la secante

Cordero, Hueso, y Torregrosa (2006) indican que en el método de Newton se necesitan realizar evaluaciones funcionales tanto de la función como de su derivada en cada paso, en ocasiones podría resultar un poco complicada la

obtención de la derivada de la función. El método de la secante no necesita la derivada para el cálculo de la estimación, pero a diferencia del método de Newton su orden de convergencia es menor, la función de punto fijo a utilizarse es:

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}}; \quad n = 0, 1, 2, \dots \quad (8)$$

El método de la secante tiene las siguientes características:

- Necesita dos estimaciones iniciales.
- Es un método con memoria.
- Es un método libre de derivadas.
- Es un método de un paso.
- El orden de convergencia aproximado es 1,618033... (Mathews & Fink, 2000).
- Al ser un método con memoria no se puede analizar si es óptimo con la conjetura de Kung y Traub.

F. Método de Steffensen

Steffensen (1933) menciona que partiendo del método de Newton descrito en la ecuación (7) y sustituyendo la derivada $f'(x_n)$ por su diferencia finita progresiva se obtiene un método con orden de convergencia cuadrática y cuya expresión es la siguiente:

$$x_{n+1} = x_n - \frac{f(x_n)^2}{f(x_n + f(x_n)) - f(x_n)}; \quad n = 0, 1, 2, \dots \quad (9)$$

García (2017) menciona algunas de las características del método de Steffensen como son:

- Es un método sin memoria.
- Es un método de un paso.
- Es un método libre de derivadas.
- Tiene un orden de convergencia cuadrática
- Según la conjetura de Kung y Traub es un método óptimo

G. Métodos Multipaso de orden alto

Con el objetivo de encontrar métodos iterativos que tengan un mayor orden de convergencia, surgen métodos que requieren, en forma general, un aumento en el número de evaluaciones funcionales, así como del número de pasos, obteniéndose de esta forma métodos multipaso cuya forma es la expresada en la ecuación (4). A continuación, se presentan dos métodos iterativos multipaso que se van a utilizar en esta investigación.

El método de Traub (1964) también conocido como el método de Potra-Pták es un método de tercer orden que se obtiene al realizar una composición entre el método de Newton cuya expresión se encuentra en la ecuación (7) consigo mismo y realizando una sustitución en el segundo paso por la derivada empleada en el primer paso, su esquema es el siguiente:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} &= y_n - \frac{f(y_n)}{f'(x_n)}; \quad n = 0, 1, 2, \dots \end{aligned} \quad (10)$$

Como menciona García (2017) el método de Traub no es óptimo.

Ostrowski (1960) Diseñó el primer método iterativo cuyo orden de convergencia es cuatro, para ello hizo uso de un esquema multipaso y, además, utilizó técnicas de interpolación, la expresión obtenida es la siguiente:

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
 x_{n+1} &= y_n - \frac{f(x_n)}{f(x_n) - 2f(y_n)} \frac{f(y_n)}{f'(x_n)}; \quad n = 0, 1, 2, \dots
 \end{aligned} \tag{11}$$

Como indica (Chicharro, 2017), el método de Ostrowski es un método óptimo.

Como indican Džunić y Petković (2012), partiendo del esquema de dos pasos de Ostrowski se puede llegar a obtener métodos iterativos de tres pasos de orden ocho. El método óptimo de orden ocho propuesto por los mismos es el siguiente:

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
 z_n &= y_n - \frac{f(x_n)}{f(x_n) - 2f(y_n)} \frac{f(y_n)}{f'(x_n)} \\
 x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \frac{[1 + f(z_n)/f(y_n)][1 + 2f(z_n)/f'(x_n)]}{[1 - 2f(y_n)/f(x_n) - (f(y_n)/f'(x_n))^2]}; \quad n = 0, 1, 2, \dots
 \end{aligned} \tag{12}$$

Cordero, Torregrosa y Vassileva (2011) diseñan un método iterativo de orden ocho óptimo, partiendo del esquema de Ostrowski y cuyo esquema es el siguiente:

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
 z_n &= x_n - \frac{f(x_n)}{f'(x_n)} \frac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)} \\
 x_{n+1} &= u_n - 3 \frac{f(z_n)}{f'(x_n)} \frac{u_n - z_n}{y_n - x_n}; \quad n = 0, 1, 2, \dots,
 \end{aligned} \tag{13}$$

donde

$$u_n = z_n - \frac{f(z_n)}{f'(x_n)} \left(\frac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)} + \frac{1}{2} \frac{f(z_n)}{f(y_n) - 2f(z_n)} \right)^2$$

III. METODOLOGIA

Referentes a física, en el presente trabajo de investigación se plantean los tres problemas siguientes:

- Esfera flotando en agua
- Caída no libre de un paracaidista
- Compresión de un resorte real

Cuyo análisis lleva a la obtención de una ecuación no lineal, y cuya solución se va a obtener mediante diferentes métodos iterativos para luego realizar un análisis entre los resultados obtenidos por cada uno de estos métodos, y de esta forma poder concluir cual método responde de mejor manera ante cada uno los problemas planteados, los métodos iterativos que se van a emplear son:

- Método de la Secante
- Método de Newton
- Método de Steffensen
- Método de Traub
- Método de Ostrowski
- Método de Džunić y Petković
- Método de Cordero, Torregrosa y Vassileva

cuyas expresiones iterativas fueron descritas en la sección anterior, y cuya programación se realizó en MATLAB, los códigos se pueden revisar en anexos.

A. Proceso

A continuación, se detalla el proceso que se va a realizar para resolver cada uno de los problemas planteados:

1) Aplicación de los fundamentos físicos y matemáticos para obtener una ecuación no lineal de la forma $f(x) = 0$, donde x será la variable física en estudio.

2) Obtención de una gráfica de la función $f(x)$, para de este modo observar los cortes con el eje x , y de esta forma poder elegir una estimación inicial lo más cercana a dicho corte, esta estimación es un parámetro de entrada para los métodos iterativos programados.

3) Selección de valores para los parámetros de entrada de los métodos programados como son: la tolerancia, el número máximo de iteraciones que junto a la estimación inicial seleccionada en el paso anterior nos permitirán obtener la solución aproximada a la ecuación no lineal.

B. Esfera flotando en agua

Una esfera de densidad ρ_e y de radio r está sumergida parcialmente en agua hasta una profundidad d , calcular dicha profundidad.

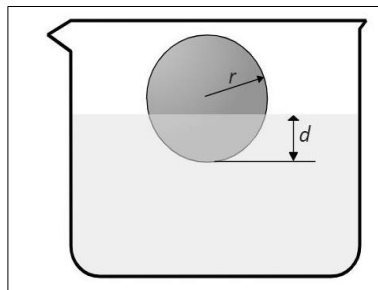


FIGURA 1. Se muestra una esfera de radio r , sumergida en agua una profundidad d .

Para calcular el volumen del casquete esférico sumergido en el agua necesitamos resolver la siguiente integral:

$$V_c = \int_0^d \pi(r^2 - (r-y)^2) dy \quad (14)$$

la cual se obtiene mediante la utilización del método de discos, donde y es la distancia desde un disco de posición arbitraria hasta el extremo del casquete esférico, y dy representa el ancho infinitesimal de dicho disco. Además, la fórmula $r^2 - (r-y)^2$ obtenida mediante el uso del teorema de Pitágoras, nos permite calcular el radio al cuadrado del disco.

Al resolver la integral planteada en la ecuación (14), se obtiene el siguiente volumen del casquete esférico sumergido en el agua:

$$V_c = \frac{\pi d^2}{3} (3r - d), \quad (15)$$

de esta forma el peso del agua desplazada es:

$$P_a = g\rho_a \frac{\pi d^2}{3} (3r - d), \quad (16)$$

donde ρ_a representa la densidad del fluido, en este caso agua.

Por otro lado, la masa de la esfera se puede calcular con la siguiente formula:

$$m_e = \rho_e \frac{4\pi r^3}{3}. \quad (17)$$

Como la esfera se encuentra parcialmente sumergida en el agua, sobre la superficie de la misma actúan los efectos de presión tanto del aire como del agua, dichos efectos aparecen en forma de fuerzas perpendiculares a su superficie. Si se analiza la flotación de la esfera en el agua se obtiene que el empuje experimentado por la esfera tiene que ser igual al peso de la misma, ya que esta se encuentra en equilibrio. De esta forma se puede calcular el empuje con la siguiente fórmula:

$$E = m_e g = g \rho_e \frac{4\pi r^3}{3}. \quad (18)$$

Aplicando el principio de Arquímedes el cual establece que "Un cuerpo total o parcialmente sumergido en un fluido en reposo experimenta un empuje vertical hacia arriba igual al peso del fluido desalojado". De esta forma igualando las ecuaciones (16) y (18), obtenemos la siguiente expresión:

$$g \rho_e \frac{4\pi r^3}{3} = g \rho_a \frac{\pi d^2}{3} (3r - d), \quad (19)$$

igualando a cero la expresión anterior:

$$\rho_a d^3 - 3r \rho_a d^2 + 4\rho_e r^3 = 0, \quad (20)$$

se obtiene una ecuación no lineal de la forma $f(d) = 0$, para continuar con la solución se va a considerar los siguientes datos:

Densidad del agua: $\rho_a = 1 \text{ g/cm}^3$

Radio de la esfera: $r = 10 \text{ cm}$

Esfera de madera de densidad: $\rho_e = 0.638 \text{ g/cm}^3$

Utilizando la función $f(d) = \rho_a d^3 - 3r \rho_a d^2 + 4\rho_e r^3$ y los respectivos datos considerados anteriormente, se obtiene la figura 2:

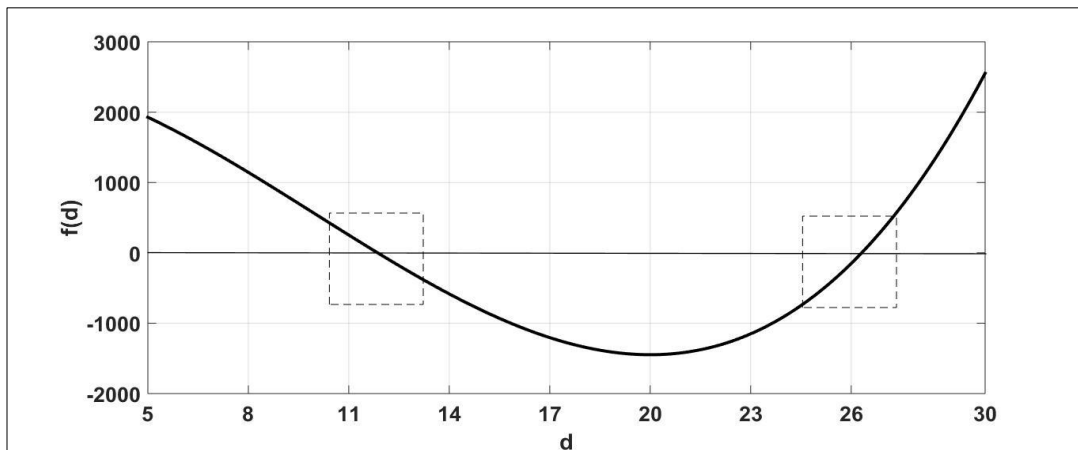


FIGURA 2. Representación gráfica de la función $f(d) = \rho_a d^3 - 3r \rho_a d^2 + 4\rho_e r^3$, para los valores de los parámetros $\rho_a = 1 \text{ g/cm}^3$, $r = 10 \text{ cm}$, $\rho_e = 0.638 \text{ g/cm}^3$.

Como se puede apreciar en la figura 2, existe dos ceros de la función en el intervalo mostrado, tomando en consideración que la esfera está parcialmente sumergida, se va a calcular solo el valor de la raíz que está del lado izquierdo, y cuyo valor está en torno a $d = 11 \text{ cm}$. Esta figura es de mucha utilidad para establecer el valor de la estimación inicial necesaria en los diferentes métodos iterativos.

Para resolver la ecuación no lineal (20) escogiendo una estimación inicial de la profundidad $d_0 = 11$ y una tolerancia 10^{-10} , para los valores de los parámetros $\rho_a = 1 \text{ g/cm}^3$, $r = 10 \text{ cm}$, $\rho_e = 0.638 \text{ g/cm}^3$ con el método de Ostrowski obtenemos la siguiente profundidad aproximada $d \approx 11.8615 \text{ cm}$, luego de realizar 3 iteraciones.

B. Caída no libre de un paracaidista

Calcular la masa m de un paracaidista que tiene una caída no libre y tiene una rapidez v_0 cuando $t=0$. Para simplificar la modelización matemática se va a despreciar el efecto de la fuerza de rozamiento vorticoso misma que es proporcional al cuadrado de la rapidez del paracaidista, para solo considerar como fuerza de resistencia la referente al rozamiento viscoso que ofrece el aire al paracaidista la cual es proporcional a la rapidez del mismo. Otros datos necesarios para el cálculo de la masa del paracaidista son el instante de tiempo $t > 0$ en donde adquiere la rapidez v misma que debe ser menor a la rapidez límite del paracaidista que es cuando el mismo desciende con fuerzas en equilibrio.

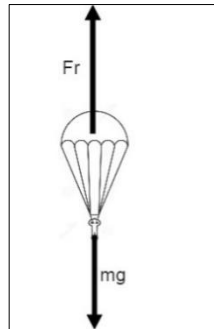


FIGURA 3. Fuerzas que actúan sobre el paracaidista durante su caída, las mismas que son fuerza de resistencia Fr dirigida hacia arriba y el peso $W = mg$ dirigido hacia abajo.

Como la fuerza de rozamiento es proporcional a la rapidez del paracaidista entonces, su módulo es:

$$Fr = kv, \quad (21)$$

Considerando la dirección del movimiento positiva hacia abajo haciendo uso de la figura 3 y aplicando la segunda ley de Newton, obtenemos:

$$W - Fr = m \frac{dv}{dt}, \quad (22)$$

en donde al reemplazar la ecuación (21) y el equivalente del peso $W = mg$, nos encontramos con la siguiente ecuación diferencial:

$$mg - kv = m \frac{dv}{dt}. \quad (23)$$

Al resolver por varias separables la ecuación (23), se plantea:

$$dt = m \frac{dv}{mg - kv}, \quad (24)$$

como tiene una rapidez v_0 cuando $t=0$, entonces la condición inicial es que $v(t=0) = v_0$, así que obtenemos:

$$\int_0^t dt = \int_{v_0}^v m \frac{dv}{mg - kv}, \quad (25)$$

al integrar los dos miembros de la ecuación:

$$t = -\frac{m}{k} \ln \left| \frac{mg - kv}{mg - kv_0} \right|, \quad (26)$$

si despejamos v , de la ecuación (26), encontramos la fórmula que relaciona la rapidez en función del tiempo, así:

$$v = \frac{mg}{k} + \left(v_0 - \frac{mg}{k} e^{-\frac{k}{m}t} \right). \quad (27)$$

Como el objetivo es encontrar la masa del paracaidista, supuesto conocido la rapidez en un instante, igualando a cero la expresión anterior:

$$\frac{mg}{k} + \left(v_0 - \frac{mg}{k} e^{-\frac{k}{m}t} \right) - v = 0, \quad (28)$$

se obtiene una ecuación no lineal de la forma $f(m) = 0$, para continuar con la solución se va a considerar los siguientes datos:

Rapidez inicial: $v_0 = 1 \text{ m/s}$

Gravedad: $g = 9.8 \text{ m/s}^2$

Constante de proporcionalidad: $k = 14 \text{ kg/s}$

Rapidez: $v = 9.8 \text{ m/s}$

Tiempo: $t = 8 \text{ s}$

Utilizando la función $f(m) = \frac{mg}{k} + \left(v_0 - \frac{mg}{k} e^{-\frac{k}{m}t} \right) - v$ y los respectivos datos considerados anteriormente, se obtiene la figura 4:

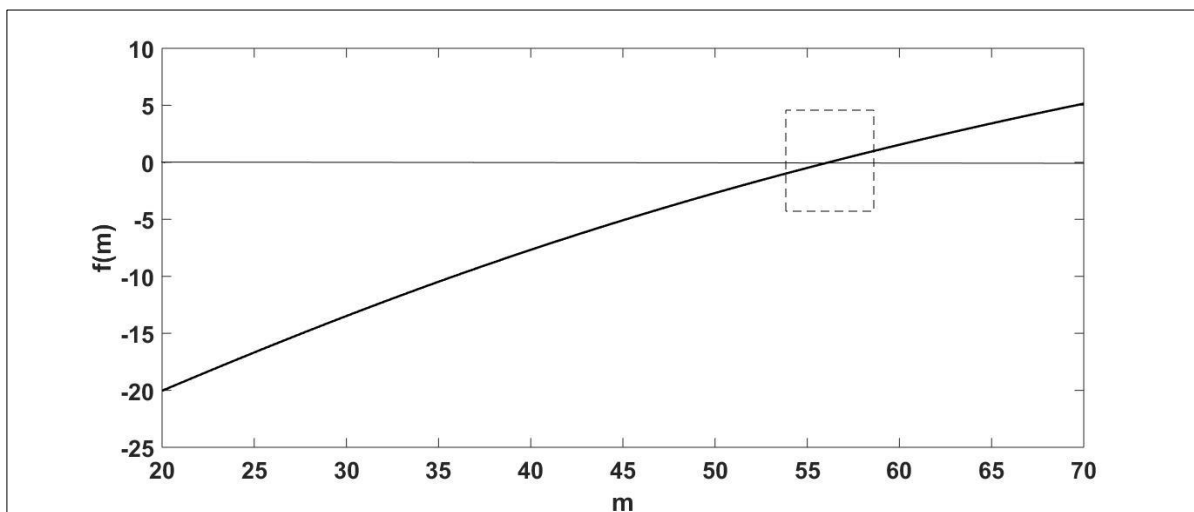


FIGURA 4. Representación gráfica de la función $f(m) = \frac{mg}{k} + \left(v_0 - \frac{mg}{k} e^{-\frac{k}{m}t} \right) - v$, para los valores de los parámetros $v_0 = 1 \text{ m/s}$, $g = 9.8 \text{ m/s}^2$, $k = 14 \text{ kg/s}$, $v = 9.8 \text{ m/s}$, $t = 8 \text{ s}$.

Como se puede apreciar en la figura anterior, existe un cero de la función en el intervalo mostrado, y cuyo valor está en torno al valor de $m = 55 \text{ kg}$, esta figura es de mucha utilidad para establecer el valor de la estimación inicial necesaria en los diferentes métodos iterativos.

Para resolver la ecuación no lineal (28) escogiendo una estimación inicial de la masa $m_0 = 60$ y una tolerancia 10^{-10} , para los valores de los parámetros $v_0 = 1 \text{ m/s}$, $g = 9.8 \text{ m/s}^2$, $k = 14 \text{ kg/s}$, $v = 9.8 \text{ m/s}$, $t = 8 \text{ s}$ con el método de Traub obtenemos la siguiente masa aproximada $m \approx 56.1698 \text{ kg}$, luego de realizar 15 iteraciones.

C. Compresión de un resorte real

En los sistemas mecánicos reales se utiliza resortes no lineales, como ejemplo de aplicación se presenta el siguiente problema: un objeto de masa m se deja caer desde una altura h sobre un resorte real cuya fuerza elástica es $F_e = -(k_1d + k_2d^{3/2})$ donde d es la compresión del mismo; calcular la máxima compresión del resorte.

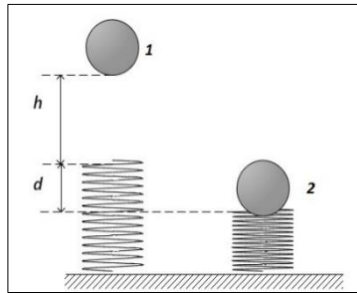


FIGURA 5. Se muestra un objeto de masa m que cae desde una altura h sobre un resorte real, a la izquierda el resorte está en posición normal, a la derecha el resorte está comprimido al máximo, d es la máxima compresión del resorte.

Utilizando el principio del Trabajo y Energía:

$$EC_1 + W_{1 \rightarrow 2} = EC_2, \tag{29}$$

como el objeto se deja caer del reposo y además se busca la máxima compresión del resorte, entonces la energía cinética del objeto es nula en las posiciones 1 y 2, además reemplazando el trabajo del peso y del resorte, tenemos:

$$W\Delta y - \int_0^d (k_1d + k_2d^{3/2})dd = 0, \tag{30}$$

la variación de y se calcula como $\Delta y = h + d$ además integrando el segundo término de la ecuación (27), obtenemos:

$$mgh + mgd - \frac{1}{2}k_1d^2 - \frac{2}{5}k_2d^{5/2} = 0, \tag{31}$$

se obtiene una ecuación no lineal de la forma $f(d) = 0$, para continuar con la solución se va a considerar los siguientes datos:

Gravedad: $g = 9.81 \text{ m/s}^2$

Constantes de proporcionalidad: $k_1 = 40000 \text{ g/s}^2$, $k_2 = 40 \text{ g/(s}^2\text{m}^{0.5})$

Masa del objeto : $m = 95 \text{ g}$

Altura: $h = 0.43 \text{ m}$

Utilizando la función $f(d) = mgh + mgd - \frac{1}{2}k_1d^2 - \frac{2}{5}k_2d^{5/2}$ y los respectivos datos considerados anteriormente, se obtiene la siguiente figura:

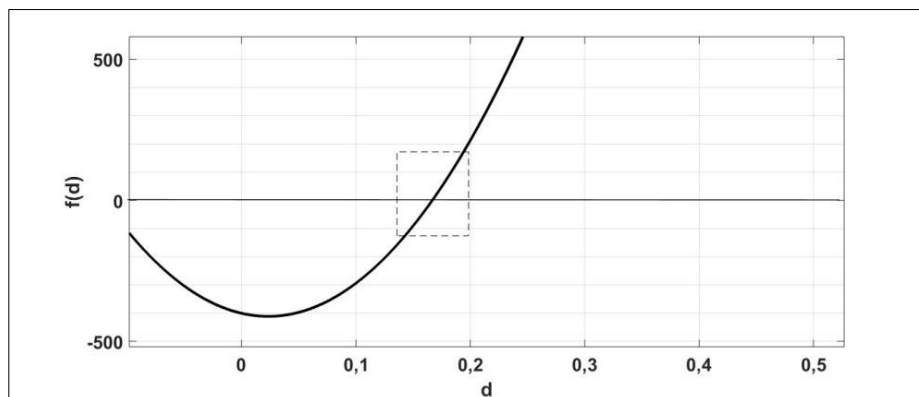


FIGURA 6. Representación gráfica de la función $f(d) = mgh + mgd - \frac{1}{2}k_1d^2 - \frac{2}{5}k_2d^{5/2}$, para los valores de los parámetros $g = 9.81 \text{ m/s}^2$, $k_1 = 40000 \text{ g/s}^2$, $k_2 = 40 \text{ g/(s}^2\text{m}^{0.5}\text{)}$, $m = 95 \text{ g}$, $h = 0.43 \text{ m}$.

Como se puede apreciar en la figura anterior, existe un cero de la función en el intervalo mostrado, y cuyo valor está en torno al valor de $d = 0.2 m$, esta figura es de mucha utilidad para establecer el valor de la estimación inicial necesaria en los diferentes métodos iterativos.

Para resolver la ecuación no lineal (31) escogiendo dos estimaciones iniciales de la máxima compresión $d_1 = 0.2$, $d_2 = 0.1$ y una tolerancia 10^{-10} , para los valores de los parámetros $g = 9.81 m/s^2$, $k_1 = 40000 g/s^2$, $k_2 = 40 g/(s^2 m^{0.5})$, $m = 95 g$, con el método de la Secante obtenemos la siguiente máxima compresión aproximada $d \approx 0.1667 m$, luego de realizar 6 iteraciones.

IV. RESULTADOS

Para las pruebas respectivas se ha utilizado MATLAB R2015a, el sistema computacional utilizado para realizar todas las pruebas tiene las siguientes especificaciones: Intel(R) i5-420U CPU @ 1.70GHz 2.40GHz, memoria RAM de 6 GB.

Los parámetros de entrada necesarios para los programas de los métodos iterativos utilizados en la presente investigación son: la función respectiva de cada uno de los problemas físicos analizados, la estimación inicial x_0 de la raíz buscada, la tolerancia que además sirve como criterio de parada y cuya representación es $|x_{n+1} - x_n| < 10^{-10}$, es decir, estamos trabajando con una tolerancia de 10^{-10} , otro parámetro de entrada es el número máximo de iteraciones, ya que se podría producir un desborde de iteraciones al no alcanzar la tolerancia establecida; dichos parámetros de entrada se los puede apreciar en los códigos de cada programa presentados en anexos.

TABLA I. Resultados obtenidos al utilizar los métodos de la Secante, Newton, Steffensen, Traub y Ostrowski para resolver tres problemas físicos la esfera flotando en agua, la caída no libre de un paracaidista y la compresión de un resorte real.

	Método	p	ξ	$ x_{n+1} - x_n $	Iter	Tiempo
Esfera flotando en agua $\rho_a d^3 - 3r\rho_a d^2 + 4\rho_e r^3 = 0$ $d_0 = 11$ $d_1^* = 11$ $d_2^* = 14$	Secante	1.6180	11.8615	1.776E-15	6	0.00552200
	Newton	2	11.8615	6.573E-14	4	0.00534150
	Steffensen	2	---	---	---	---
	Traub	3	11.8615	1.066E-14	3	0.00527680
	Ostrowski	4	11.8615	1.776E-15	3	0,00534590
	Džunić y Petković	8	11.8615	8.882E-15	3	0.00447070
	Cordero, Torregrosa y Vassileva	8	11.8615	1.776E-15	3	0.00441745
Caída no libre de un paracaidista $\frac{mg}{k} \left(1 - e^{-\frac{m}{k}t} \right) - v = 0$ $m_0 = 60$ $m_1^* = 55$ $m_2^* = 60$	Secante	1.6180	56.1698	3.2188E-12	5	0.00483015
	Newton	2	56.1698	6.1036E-11	28	0.00478300
	Steffensen	2	---	---	---	---
	Traub	3	56.1698	3.4710E-11	15	0.00476565
	Ostrowski	4	---	---	---	---
	Džunić y Petković	8	---	---	---	---
	Cordero, Torregrosa y Vassileva	8	---	---	---	---
Compresión de un resorte real $mgh + mgd - \frac{1}{2}k_1 d^2 - \frac{2}{5}k_2 d^{5/2} = 0$ $d_0 = 0.2$ $d_1^* = 0.1$ $d_2^* = 0.2$	Secante	1.6180	0.1667	1.002E-11	6	0.00536435
	Newton	2	0.1667	2.776E-17	5	0.00510065
	Steffensen	2	0.1667	6.500E-12	682	0.01932860
	Traub	3	0.1667	0.000E+00	4	0.00524890
	Ostrowski	4	0.1667	2.776E-17	3	0.00491215
	Džunić y Petković	8	---	---	---	---
	Cordero, Torregrosa y Vassileva	8	---	---	---	---

* Estimaciones iniciales para el caso del método de la Secante.

En la tabla I se muestran los resultados obtenidos al emplear los métodos iterativos de la Secante, Newton, Steffensen, Traub, Ostrowski, Džunić y Petković y Cordero, Torregrosa y Vassileva; para resolver tres problemas físicos; encontramos la estimación inicial x_0 utilizada para alcanzar la raíz ξ , el orden de convergencia p , el error cometido en la última iteración $|x_{n+1} - x_n|$, el número de iteraciones (Iter) necesarias para cumplir con la tolerancia del criterio de parada, así como el tiempo necesario en cómputo (tiempo) el cual se calculó mediante el promedio de 20 ejecuciones consecutivas de cada programa.

Para los parámetros de entrada como son la tolerancia menor a 10^{-10} y las respectivas estimaciones iniciales de la raíz para cada problema físico, se producen problemas con la utilización de algunos métodos ya que no son capaces de resolver las diferentes ecuaciones no lineales utilizadas; como ejemplo el método de Steffensen no consigue la solución del problema de la esfera flotando en agua, los dos métodos de orden ocho no consiguen resolver los problemas de la caída no libre de un paracaidista y el de la compresión real de un resorte.

Para el caso de la esfera flotando en agua, tomando como criterios de análisis el tiempo computacional requerido, el número de iteraciones y el error, se observa que el método de orden 8 de Cordero, Torregrosa y Vassileva es el mejor de todos los métodos utilizados, recordando además que es un método óptimo.

Para el caso de la caída no libre de un paracaidista, tomando como criterios de análisis el tiempo computacional requerido, el número de iteraciones y el error, se observa que el método de orden 3 de Traub es el mejor de todos los métodos utilizados, recordando además que es un método no óptimo.

Finalmente, para el caso de la compresión de un resorte real, tomando como criterios de análisis el tiempo computacional requerido, el número de iteraciones y el error, se observa que el método de orden 3 de Traub es el mejor de todos los métodos utilizados, recordando además que es un método no óptimo.

V. CONCLUSIONES

Si bien al momento de resolver una ecuación no lineal el método por excelencia a ser utilizado es Newton, en el presente trabajo se ha demostrado que existen otros métodos que ofrecen mejores características ya sea por la estructura de los mismos como es el caso de aquellos métodos libres de derivadas, así como por el orden de convergencia alto.

Al utilizar métodos iterativos con mayor orden de convergencia se reduce el número de iteraciones requeridas en la obtención de la solución aproximada de la ecuación no lineal estudiada, reduciendo de esta forma el tiempo de ejecución siendo más eficientes que los métodos tradicionales de segundo orden como son Newton o Steffensen.

Para los problemas planteados el método de Newton siempre converge a la solución, convirtiéndose de esta forma en el método más estable empleado en nuestra investigación.

Al utilizar funciones un tanto más complejas como en el caso de esta investigación la caída no libre de un paracaidista que tiene una naturaleza exponencial, y la compresión real de un resorte que tiene términos irracionales los métodos de orden ocho utilizados en esta investigación no siempre convergen a la solución, por lo tanto, son métodos inestables.

Para entender de mejor forma los criterios de estabilidad de un método que hacen que algunos métodos no convergen a la solución, es necesario realizar un estudio dinámico de los mismos, para ello existe diferentes investigaciones que pueden ser revisadas con este propósito.

Los métodos iterativos de punto fijo tienen una amplia gama de aplicación en diferentes campos del conocimiento, en todos aquellos en donde la modelación matemática lleve a la obtención de ecuaciones no lineales.

En próximas investigaciones se podría mejorar los fenómenos físicos estudiados, como ejemplo para la caída no libre del paracaidista se podría utilizar un modelo más real para la fuerza de rozamiento, en donde la misma sea proporcional a la rapidez al cuadrado.

REFERENCIAS

Amorós, C. (2020). *Estudio sobre convergencia y dinámica de los métodos de Newton, Stirling y alto orden*. Madrid: Universidad Internacional de la Rioja.

Artidiello, S. (2014). *Diseño, implementación y convergencia de métodos iterativos para resolver ecuaciones y sistemas no lineales utilizando funciones peso*. Valencia: Universitat Politècnica de Valencia.

- Chicharro, F. (2017). *Análisis dinámico y aplicaciones de métodos iterativos de resolución de ecuaciones no lineales*. Valencia: Universitat Politècnica de Valencia.
- Cordero, A., Hueso, J., & Torregrosa, J. (2006). *Problemas resueltos de métodos numéricos*. Madrid: Paraninfo.
- Cordero, A., Torregrosa, J. R., & Vassileva, M. P. (2011). Three-step iterative methods with optimal eighth-order convergence. *Journal of Computational and Applied Mathematics*, 235, 3189-3194.
- Džunić, J., & Petković, M. S. (2012). A Family of Three-Point Methods of Ostrowski's Type for Solving Nonlinear Equations. *Journal of Applied Mathematics*.
- García, J. (2017). *Análisis dinámico*. Valencia: Universidad Politècnica de Valencia.
- Kung, H. T., & Traub, J. F. (1974). Optimal order of one-point and multipoint iteration. *J. Assoc. Comput. Math*, 21(4), 643-651.
- Mathews, J. H., & Fink, K. D. (2000). *Métodos numéricos con Matlab*. Madrid: Prentice Hall.
- Ostrowski, A. M. (1960). *Solution of equations and systems of equations*. New York: Academic Press.
- Ostrowski, A. M. (1966). *Solutions of equations and systems of equations*. Academic Press.
- Steffensen, J. F. (1933). Remarks on iteration. *Skand. Aktuarietidskrift*, 64-72.
- Traub, J. F. (1964). *Iterative methods for the solution of equations*. Englewood Cliffs, N.J., Prentice-Hall.
- Ypma, T. J. (1995). Historical development of the Newton-Raphson method. *SIAM Rev.*, 37, 531-551.

ANEXO 1: Método de la Secante

```

function [x,iter,incre1,incre2,p ] = Secante(funcion,x0,x1,tol,maxiter)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
%inicializacion de variables
tic
iter=0;
incre1=tol+1;% error consecutivo entre dos iterados
incre2=incre1;% error ultimo iterado
Inc=[];
while iter<maxiter && incre1>tol && incre2>tol% dos posibles criterios en cuanto al error
    f=feval(funcion,x0);
    f1=feval(funcion, x1);
    x=x1-f1*(x1-x0)/(f1-f);
    iter=iter+1;
    incre1=abs(x-x1);
    incre2=abs(f);
    Inc=[Inc,incre1];%
    x0=x1;
    x1=x;
end
iter
incre1
% para el ACOC
p1=log(Inc(3:end)./Inc(2:end-1));
p2=log(Inc(2:end-1)./Inc(1:end-2));
p=p1./p2;
plot(p)
if incre1>tol;
    disp('Necesita mas iteraciones');
end
toc
end
function [y,dy]=esfera(x)
r=10;
d=0.638;
y=x^3-3*x^2*r+4*r^3*d;
dy=3*x^2-6*x*r;
end
function [y,dy]=paracaidista(x)
g=9.81;
c=14;
v=35;
t=8;
y=g*x/c-g*x/c*exp(-c*t/x)-v;
dy=g/c-g/c*exp(-c*t/x)*c*x^-2*t;
end
function [y,dy]=resorte(x)
k1=40000;
k2=40;
m=95;
g=9.81;
h=0.43;
y=2/5*k2*x^(5/2)+1/2*k1*x^2-m*g*x-m*g*h;
dy=k2*x^(3/2)+k1*x-m*g;
end

```

ANEXO 2: Método de Newton

```

function [x,iter,incre1,incre2,p ] = Newton(funcion,x0,tol,maxiter)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
%inicializacion de variables
tic
iter=0;
incre1=tol+1;% error consecutivo entre dos iterados
incre2=incre1;% error ultimo iterado
Inc=[];
while iter<maxiter && incre1>tol && incre2>tol% dos posibles criterios en cuanto al error
    [f,df]=feval(funcion,x0);
    x=x0-f/df;
    iter=iter+1;
    incre1=abs(x-x0);
    incre2=abs(f);
    Inc=[Inc,incre1];%
    x0=x;
end
iter
incre1
% para el ACOC
p1=log(Inc(3:end)./Inc(2:end-1));
p2=log(Inc(2:end-1)./Inc(1:end-2));
p=p1./p2;
plot(p)
if incre1>tol;
    disp('Necesita mas iteraciones');
end
toc
end
function [y,dy]=esfera(x)
r=10;
d=0.638;
y=x^3-3*x^2*r+4*r^3*d;
dy=3*x^2-6*x*r;
end
function [y,dy]=paracaidista(x)
g=9.81;
c=14;
v=35;
t=8;
y=g*x/c-g*x/c*exp(-c*t/x)-v;
dy=g/c-g/c*exp(-c*t/x)*c*x^-2*t;
end
function [y,dy]=resorte(x)
k1=40000;
k2=40;
m=95;
g=9.81;
h=0.43;
y=2/5*k2*x^(5/2)+1/2*k1*x^2-m*g*x-m*g*h;
dy=k2*x^(3/2)+k1*x-m*g;
end

```


ANEXO 3: Método de Steffensen

```

function [x,iter,incre1,incre2,p ] = Steffensen(funcion,x0,tol,maxiter)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
%inicializacion de variables
tic
iter=0;
incre1=tol+1;% error consecutivo entre dos iterados
incre2=incre1;% error ultimo iterado
Inc=[];
while iter<maxiter && incre1>tol && incre2>tol% dos posibles criterios en cuanto al error
    f=feval(funcion,x0);
    w=x0+f;
    fw=feval(funcion,w);
    x=x0-f^2/(fw-f);
    iter=iter+1;
    incre1=abs(x-x0);
    incre2=abs(f);
    Inc=[Inc,incre1];%
    x0=x;
end
iter
incre1
% para el ACOC
p1=log(Inc(3:end)./Inc(2:end-1));
p2=log(Inc(2:end-1)./Inc(1:end-2));
p=p1./p2;
plot(p)
if incre1>tol;
    disp('Necesita mas iteraciones');
end
toc
end
function [y,dy]=esfera(x)
r=10;
d=0.638;
y=x^3-3*x^2*r+4*r^3*d;
dy=3*x^2-6*x*r;
end
function [y,dy]=paracaidista(x)
g=9.81;
c=14;
v=35;
t=8;
y=g*x/c-g*x/c*exp(-c*t/x)-v;
dy=g/c-g/c*exp(-c*t/x)*c*x^-2*t;
end
function [y,dy]=resorte(x)
k1=40000;
k2=40;
m=95;
g=9.81;
h=0.43;
y=2/5*k2*x^(5/2)+1/2*k1*x^2-m*g*x-m*g*h;
dy=k2*x^(3/2)+k1*x-m*g;
end

```

ANEXO 4: Método de Traub

```

function [x,iter,incre1,incre2,p ] = Traub(funcion,x0,tol,maxiter)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
%inicializacion de variables
tic
iter=0;
incre1=tol+1;% error consecutivo entre dos iterados
incre2=incre1;% error ultimo iterado
Inc=[];
while iter<maxiter && incre1>tol && incre2>tol% dos posibles criterios en cuanto al error
[f,df]=feval(funcion,x0);
y=x0-f/df;
fy=feval(funcion,y);
x=y-fy/df;
iter=iter+1;
incre1=abs(x-x0);
incre2=abs(f);
Inc=[Inc,incre1];%
x0=x;
end
iter
incre1
% para el ACOC
p1=log(Inc(3:end)./Inc(2:end-1));
p2=log(Inc(2:end-1)./Inc(1:end-2));
p=p1./p2;
plot(p)
if incre1>tol;
disp('Necesita mas iteraciones');
end
toc
end
function [y,dy]=esfera(x)
r=10;
d=0.638;
y=x^3-3*x^2*r+4*r^3*d;
dy=3*x^2-6*x*r;
end
function [y,dy]=paracaidista(x)
g=9.81;
c=14;
v=35;
t=8;
y=g*x/c-g*x/c*exp(-c*t/x)-v;
dy=g/c-g/c*exp(-c*t/x)*c*x^-2*t;
end
function [y,dy]=resorte(x)
k1=40000;
k2=40;
m=95;
g=9.81;
h=0.43;
y=2/5*k2*x^(5/2)+1/2*k1*x^2-m*g*x-m*g*h;
dy=k2*x^(3/2)+k1*x-m*g;
end

```

ANEXO 5: Método de Ostrowski

```

function [x,iter,incre1,incre2,p ] = Ostrowski(funcion,x0,tol,maxiter)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
%inicializacion de variables
tic
iter=0;
incre1=tol+1;% error consecutivo entre dos iterados
incre2=incre1;% error ultimo iterado
Inc=[];
while iter<maxiter && incre1>tol && incre2>tol% dos posibles criterios en cuanto al error
    [f,df]=feval(funcion,x0);
    y=x0-f/df;
    fy=feval(funcion,y);
    x=y-f/(f-2*fy)*fy/df;
    iter=iter+1;
    incre1=abs(x-x0);
    incre2=abs(f);
    Inc=[Inc,incre1];%
    x0=x;
end
iter
incre1
% para el ACOC
p1=log(Inc(3:end)./Inc(2:end-1));
p2=log(Inc(2:end-1)./Inc(1:end-2));
p=p1./p2;
plot(p)
if incre1>tol;
    disp('Necesita mas iteraciones');
end
toc
end
function [y,dy]=esfera(x)
r=10;
d=0.638;
y=x^3-3*x^2*r+4*r^3*d;
dy=3*x^2-6*x*r;
end
function [y,dy]=paracaidista(x)
g=9.81;
c=14;
v=35;
t=8;
y=g*x/c-g*x/c*exp(-c*t/x)-v;
dy=g/c-g/c*exp(-c*t/x)*c*x^-2*t;
end
function [y,dy]=resorte(x)
k1=40000;
k2=40;
m=95;
g=9.81;
h=0.43;
y=2/5*k2*x^(5/2)+1/2*k1*x^2-m*g*x-m*g*h;
dy=k2*x^(3/2)+k1*x-m*g;
end

```

ANEXO 6: Método de Džunić y Petković

```

function [x,iter,incre1,incre2,p ] = Dzunie(funcion,x0,tol,maxiter)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
%inicializacion de variables
tic
iter=0;
incre1=tol+1;% error consecutivo entre dos iterados
incre2=incre1;% error ultimo iterado
Inc=[];
while iter<maxiter && incre1>tol && incre2>tol% dos posibles criterios en cuanto al error
[f,df]=feval(funcion,x0);
y=x0-f/df;
fy=feval(funcion,y);
z=y-f/(f-2*fy)*fy/df;
fz=feval(funcion,z);
x=z-(1+fz/fy)*(1+2*fz/f)/(1-2*fy/f-(fy/f)^2)*fz/df;
iter=iter+1;
incre1=abs(x-x0);
incre2=abs(f);
Inc=[Inc,incre1];%
x0=x;
end
iter
incre1
% para el ACOG
p1=log(Inc(3:end)./Inc(2:end-1));
p2=log(Inc(2:end-1)./Inc(1:end-2));
p=p1./p2;
plot(p)
if incre1>tol;
disp('Necesita mas iteraciones');
end
toc
end
function [y,dy]=esfera(x)
r=10;
d=0.638;
y=x^3-3*x^2*r+4*r^3*d;
dy=3*x^2-6*x*r;
end
function [y,dy]=paracaidista(x)
g=9.81;
c=14;
v=35;
t=8;
y=g*x/c-g*x/c*exp(-c*t/x)-v;
dy=g/c-g/c*exp(-c*t/x)*c*x^-2*t;
end
function [y,dy]=resorte(x)
k1=40000;
k2=40;
m=95;
g=9.81;
h=0.43;
y=2/5*k2*x^(5/2)+1/2*k1*x^2-m*g*x-m*g*h;
dy=k2*x^(3/2)+k1*x-m*g;
end

```

ANEXO 7: Método de Cordero, Torregrosa y Vassileva

```

function [x,iter,incre1,incre2,p ] = M8(funcion,x0,tol,maxiter)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
%inicializacion de variables
tic
iter=0;
incre1=tol+1;% error consecutivo entre dos iterados
incre2=incre1;% error ultimo iterado
Inc=[];
while iter<maxiter && incre1>tol && incre2>tol% dos posibles criterios en cuanto al error
    [f,df]=feval(funcion,x0);
    y=x0-f/df;
    fy=feval(funcion,y);
    z=x0-(f-fy)/(f-2*fy)*f/df;
    fz=feval(funcion,z);
    u=z-fz/df*((f-fy)/(f-2*fy)+1/2*fz/(fy-2*fz))^2;
    x=u-3*fz/df*(u-z)/(y-x0);
    iter=iter+1;
    incre1=abs(x-x0);
    incre2=abs(f);
    Inc=[Inc,incre1];%
    x0=x;
end
iter
incre1
% para el ACOC
p1=log(Inc(3:end)./Inc(2:end-1));
p2=log(Inc(2:end-1)./Inc(1:end-2));
p=p1./p2;
plot(p)
if incre1>tol;
    disp('Necesita mas iteraciones!');
end
toc
end
function [y,dy]=esfera(x)
r=10;
d=0.638;
y=x^3-3*x^2*r+4*r^3*d;
dy=3*x^2-6*x*r;
end
function [y,dy]=paracaidista(x)
g=9.81;
c=14;
v=35;
t=8;
y=g*x/c-g*x/c*exp(-c*t/x)-v;
dy=g/c-g/c*exp(-c*t/x)*c*x^-2*t;
end
function [y,dy]=resorte(x)
k1=40000;
k2=40;
m=95;
g=9.81;
h=0.43;
y=2/5*k2*x^(5/2)+1/2*k1*x^2-m*g*x-m*g*h;
dy=k2*x^(3/2)+k1*x-m*g;
end

```