

# Premios del Departamento de Matemáticas de la Universidad Autónoma de Madrid para Estudiantes de Secundaria

**Sexta Edición, 2011/2012**

**TRABAJO:** Búsqueda de un algoritmo eficiente para la aproximación de las raíces de una ecuación

***GANADOR EN LA CATEGORÍA DE BACHILLERATO***

**AUTORES:**

- o Estela Sanz Jiménez
- o Ángel Sanz Núñez

**TUTOR:**

- o Fernando Alcaide

**CENTRO:** IES María Guerrero (Collado Villalba, Madrid)



# Búsqueda de un algoritmo eficiente para la aproximación de las raíces de una ecuación

---

## Trabajo de investigación - Bachillerato

### 1. Introducción y objetivos

En el presente trabajo de investigación se intenta aportar una solución al problema de la resolución eficiente de las ecuaciones no lineales de la forma  $f(x) = 0$ .

Para la mayoría de los casos, resulta imposible conocer el valor exacto de las raíces de una ecuación. Por ello, es necesario intentar obtener un valor que se aproxime, todo lo que interese, al valor verdadero de la raíz, para lo cual se emplean los algoritmos de aproximación de dichas raíces.

Debido a la gran capacidad de procesamiento de los ordenadores actuales, la diferencia entre emplear un algoritmo u otro no es importante. El interés del estudio de los algoritmos de aproximación de raíces es, por tanto, puramente teórico.

El interés inicial de este trabajo se centra en tres objetivos principales:

a) *Estudiar los algoritmos tradicionales más importantes, como un primer paso para entenderlos y descubrir estrategias que puedan ser útiles.*

b) *Llegar al completo desarrollo de un algoritmo efectivo en la búsqueda de raíces de una ecuación, en base a métodos ya existentes pero incluyendo alguna característica nueva.*

c) *Implementar el nuevo algoritmo en el lenguaje de programación Python con el fin de comprobar su efectividad por ordenador.*

### 2. Trasfondo histórico

El problema de la resolución de la ecuación no lineal  $f(x) = 0$  es un tema amplísimo, que ha dado qué pensar a matemáticos durante cientos de años. Uno de los ejemplos más sencillos es el caso en el que la función  $f(x)$  es un polinomio. La resolución de los polinomios de primer y segundo grado era conocida por pueblos antiguos, como los babilonios. En torno a la década de 1540, Girolamo Cardano, Niccolò Fontana Tartaglia y Scipione del Ferro desarrollaron y compitieron por la solución algebraica de la ecuación cúbica, que fue finalmente expuesta en la obra *Ars Magna* (Cardano, 1545). En ella, el autor reconoce que Tartaglia le había mostrado anteriormente su forma de resolver cierto tipo de ecuaciones cúbicas, y también afirma que Scipione del Ferro había descubierto esa misma fórmula de forma independiente y antes que Tartaglia. Cardano muestra el método para resolver *cualquier* ecuación cúbica. En la obra también se muestra el método para resolver ecuaciones cuárticas (de grado cuatro), que había sido desarrollado principalmente por Lodovico Ferrari, alumno y secretario de Cardano (aunque el método de resolución de las ecuaciones cuárticas fue más tarde desarrollado por otros matemáticos, como Descartes o Euler).

Posteriormente, muchos matemáticos trataron de dar una solución general a las ecuaciones polinómicas de grado quinto y superior, pero los esfuerzos fueron en vano. De hecho, el teorema de Abel-Ruffini, que se considera completamente probado en 1824, con la aportación de Niels

Henrik Abel, demuestra que no existe una solución algebraica *general* para ecuaciones polinómicas de quinto grado, lo que significa que no existe un método algebraico que permita resolver *todas* las ecuaciones polinómicas de quinto grado, si bien la solución exacta de algunas de ellas sí puede ser obtenida.

En este momento es cuando comienzan a cobrar importancia los algoritmos de aproximación de raíces. Puesto que no se puede obtener la solución exacta, es conveniente contar al menos con una forma de aproximarla con toda la exactitud que se desee. Ésta es la idea general de los mencionados algoritmos. Sin embargo, los métodos de aproximación de raíces ya se conocían antes de que el teorema de Abel-Ruffini fuera probado: el método de la “regula falsi” ya era conocido en el III a.C.

Del mismo modo, el método de Newton-Raphson fue descrito por Isaac Newton en *De analysi per aequationes numero terminorum infinitas* (escrito en 1669, publicado en 1711) y en *De methodis fluxionum et serierum infinitarum* (escrito en 1671, traducido y publicado en 1736). El método como tal fue publicado por primera vez en 1685 en *A Treatise of Algebra both Historical and Practical*, por John Wallis. En 1690, Joseph Raphson publicó una revisión simplificada del método en *Analysis aequationum universalis*. Si bien las bases fueron establecidas por Newton, su forma original de llevarlo a cabo difiere bastante de la actual, y él sólo lo aplicaba a ecuaciones polinómicas. En la obra mencionada de Raphson, éste hace una descripción más parecida a la actual.

### 3. Descripción y desarrollo de algunos métodos clásicos para la aproximación de las raíces de una ecuación

A continuación se exponen los distintos algoritmos que han servido como base para la búsqueda de nuevas técnicas. En todos ellos, se requiere una función  $f(x)$  y un intervalo  $[a, b]$  en el que  $f$  sea continua y se encuentre la raíz a aproximar,  $r$ .

#### 3.1 Método de bisección

Consigue acercarse a la raíz de la ecuación tanto como se quiera, haciendo divisiones iguales sucesivas en el intervalo de partida, hallando el punto medio de dicho intervalo y aplicando el teorema de Bolzano en cada iteración, eligiendo en cada paso la subdivisión del intervalo en la que se encuentre la raíz. La subdivisión se repite tantas veces como sea preciso hasta conseguir un resultado con un error menor que una cantidad prefijada (la *tolerancia*).

El algoritmo de bisección procede de la manera siguiente:

1. Se considera la función  $y = f(x)$ , que cumple las condiciones mencionadas.
2. Se toma  $c_1 = \frac{a+b}{2}$ ;  $c_1 \in (a, b)$
3. Si  $f(c_1) = 0$ , entonces  $c_1 = r$  es la raíz que se pretende encontrar, y el método finaliza aquí. En caso contrario, se continúa con el paso 4.
4. Se tiene *una* de las siguientes opciones:

$$a. \text{ signo } f(c_1) \neq \text{ signo } f(a) \Leftrightarrow \exists r \in (a, c_1) / f(r) = 0$$

$$b. \text{ signo } f(c_1) \neq \text{ signo } f(b) \Leftrightarrow \exists r \in (c_1, b) / f(r) = 0$$

5. Se toma el intervalo en el cual se cumpla  $\exists r / f(r) = 0$ .
6. Se determina el valor  $c_2 = \frac{a+c_1}{2}$  ó  $c_2 = \frac{c_1+b}{2}$ , en función el intervalo elegido en el paso anterior.
7. Si  $f(c_2) = 0$ , entonces  $c_2 = r$  es la raíz que se pretende encontrar. En caso contrario, se continúa con el paso 8.
8. Se repite el proceso de forma análoga al paso 4.

El proceso se repite un número  $n$  de veces hasta obtener un  $c_n = r$  o, lo que es más habitual, un  $c_n$  que se aproxime a la solución  $r$  lo suficiente como para considerar aceptable el error cometido en la estimación, es decir, hasta que  $|c_n - r| < T$  con una tolerancia  $T$  prefijada.

La ventaja más importante del método de bisección es que siempre converge a la solución, sea cual sea la ecuación en cuestión, siempre y cuando se verifiquen las condiciones iniciales. Sin embargo, no es un método eficiente ya que pueden ser necesarias un gran número de iteraciones para obtener una solución con un error aceptable.

### 3.2 Método de la “regula falsi”

El método de la “regula falsi” se basa en la idea de que otra forma de realizar la aproximación consiste en trazar una recta secante a la gráfica de la función, que pase por dos puntos dados, de modo que el punto de corte de esta recta secante con el eje de abscisas es una aproximación de la raíz. Los puntos iniciales requeridos por el algoritmo no son más que  $A = (a, f(a))$  y  $B = (b, f(b))$ .

De este modo, el algoritmo funciona así:

1. Se considera la función  $y = f(x)$ , que cumple las condiciones mencionadas.
2. Se determina la recta  $r_1$ , que pasa por  $A = (a, f(a))$  y  $B = (b, f(b))$ . Esta recta será secante a la gráfica de  $f(x)$ , y además cortará al eje de abscisas en un punto  $c$ . El vector director de  $r_1$  será el que une los puntos  $A$  y  $B$ , es decir

$$\overrightarrow{AB} = (b - a, f(b) - f(a))$$

Y la ecuación de la recta  $r_1$ , definida como la recta de pendiente  $m_{r_1} = \frac{f(b)-f(a)}{b-a}$  y que pasa por  $A$ , queda así:

$$r_1 \equiv y - f(a) = \frac{f(b) - f(a)}{b - a} \cdot (x - a)$$

La recta  $r_1$  corta a  $OX$  en un punto  $C = (c, 0)$ . Evidentemente, este punto pertenece a  $r_1$ , por lo que

$$0 - f(a) = \frac{f(b) - f(a)}{b - a} \cdot (c - a)$$

De donde se desarrolla  $c$ :

$$\begin{aligned} c &= -\frac{b - a}{f(b) - f(a)} \cdot f(a) + a = \frac{(a - b) \cdot f(a) + a \cdot (f(b) - f(a))}{f(b) - f(a)} \\ &= \frac{\cancel{f(a)} \cdot a - f(a) \cdot b + f(b) \cdot a - \cancel{f(a)} \cdot a}{f(b) - f(a)} = \frac{f(b) \cdot a - f(a) \cdot b}{f(b) - f(a)} \end{aligned}$$

3. Si  $c$  es una aproximación aceptable o es el valor exacto de la raíz, la tomamos sin más, en caso contrario se sigue con el paso 4.
4. Se evalúa el signo de  $f(c)$  y se tiene una de las siguientes opciones:
  - a.  $\text{signo } f(c) \neq \text{signo } f(a) \Leftrightarrow \exists r \in (a, c) / f(r) = 0$
  - b.  $\text{signo } f(c) \neq \text{signo } f(b) \Leftrightarrow \exists r \in (c, b) / f(r) = 0$
5. Se toma el intervalo en el que se cumpla  $\exists r / f(r) = 0$
6. Se repite desde el paso 2 para hallar un nuevo punto de intersección de la siguiente recta,  $r_2$ , con el eje de abscisas, comenzando así la segunda iteración.

En general, se puede decir que el método de la "regula falsi" emplea un intervalo inicial  $(a_n, b_n)$  en el que se comprueba que exista al menos una raíz por medio del Teorema de Bolzano, y a partir del cual se obtiene un punto  $c_n$  que vale

$$c_n = \frac{f(b_n) \cdot a_n - f(a_n) \cdot b_n}{f(b_n) - f(a_n)}$$

siendo  $n$  el número de iteración. A la vista del valor de  $c_n$ , o bien se toma como solución (si el error cometido se considera aceptable o si  $c_n = r$ ), o bien se toma como un extremo del intervalo de la siguiente iteración, que devolverá a su vez un valor  $c_{n+1}$  que se evaluará idénticamente. Este proceso se iterará el número de veces que se considere necesario o hasta que se halle el valor exacto de  $r$ .

### 3.3 Método de pendiente fija

Adicionalmente, este método requiere un punto inicial  $X_0 \in [a, b]$  y una pendiente, de la que se hablará más adelante. El algoritmo halla  $f(X_0)$ , y traza una recta que pasa por el punto  $(X_0, f(X_0))$  y tiene una pendiente  $m$ , y halla el punto en el que esta recta corta al eje de abscisas. Se calcula la imagen de este nuevo punto, y se traza otra recta que pasa por esa imagen y tiene la misma pendiente. Los puntos intersección de las sucesivas rectas con el eje de abscisas formarán una sucesión de valores aproximados de la solución. El algoritmo en detalle es éste:

1. Se considera la función  $y = f(x)$ , que cumple las condiciones mencionadas.

2. Se parte de una abscisa inicial  $X_0$ . Por simplicidad, se puede tomar  $X_0 = a$ . Se halla  $f(X_0)$ .
3. Se calcula la intersección de la recta  $r \equiv y = m \cdot (x - X_0) + f(X_0)$  con el eje de abscisas.

$$\begin{cases} y = m \cdot (x - X_0) + f(X_0) \\ y = 0 \end{cases} \rightarrow x = X_1 = X_0 - \frac{f(X_0)}{m}$$

4. El proceso se repite el número  $n$  deseado de veces, obteniendo, en ciertas condiciones, valores más próximos a la raíz  $r$  de la ecuación.

Nótese que también se puede comenzar por el otro extremo del intervalo, haciendo  $X_0 = b$ , siempre que se tome la pendiente con el signo que garantice la convergencia (véase a continuación).

El método se puede resumir en la siguiente sucesión, definida por recurrencia, que converge a la raíz  $r$  de la ecuación  $f(x) = 0$  :

$$\begin{cases} X_0 = a \\ X_{n+1} = X_n - \frac{f(X_n)}{m} \end{cases}$$

Este algoritmo es más cómodo de ejecutar manualmente en comparación con el método de bisección. Además, la convergencia a la raíz suele ser algo más rápida que en tal método.

El mayor problema que se presenta consiste en la elección del valor de la pendiente  $m$ . Precisamente, este es el único factor que afecta a la convergencia del método. El signo de la misma es claro, pues ha de satisfacer la condición siguiente:

$$\text{signo } m = -\text{signo } f(a) = \text{signo } f(b)$$

Se suele recomendar que  $|m|$  valga "algo más" que el valor de la expresión

$$\frac{|f(b) - f(a)|}{b - a}$$

aunque esta noción es imprecisa, por lo que a veces es conveniente probar con varios valores antes de desarrollar el algoritmo completo.

### 3.4 Método de Newton-Raphson

Es considerablemente más rápido que los tres métodos expuestos anteriormente, pero también es más restrictivo. Da una solución al mayor problema que presentaba el método anterior: el valor de la pendiente, que es el valor de  $f'(x)$  en cada punto  $X_n$ . Esto significa que se van trazando rectas *tangentes* a la función en cada punto. Por esto, se requiere una condición más: que  $f''(x)$  tenga signo constante en todo el intervalo  $(a, b)$ .

La siguiente sucesión, definida por recurrencia,

$$\begin{cases} X_0 = a \\ X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)} \end{cases}$$

converge a la raíz  $r$  de la ecuación  $f(x) = 0$  que se encuentra en el intervalo  $(a, b)$ . Como valor inicial  $X_0$  se toma el extremo del intervalo cuyo signo coincide con el signo de  $f''(x)$  en dicho intervalo. De esta forma se produce la convergencia óptima y, de hecho, de tomarse el extremo contrario, podría no producirse la convergencia.

### **Demostración:**

Ahora se expone una demostración de la convergencia del método de Newton-Raphson. Las condiciones iniciales son las siguientes:

- Función continua en  $[a, b]$  real de variable real  $y = f(x)$  definida.
- $\exists r \in (a, b) / f(r) = 0 \leftrightarrow \text{signo } f(a) = -\text{signo } f(b)$
- $\text{signo } f''(x)$  es constante  $\forall x \in [a, b]$

Lo que se busca probar es que el método converge a la raíz, es decir, que sirva para aproximarla. Esto significa que lo que se ha de probar es que  $a < X_1 < r$  o que  $r < X_1 < a$ . Por tanto, el objetivo es llegar a una expresión en la que se puedan relacionar algunos de esos elementos.

Para hacer la demostración más sencilla, y sin que esto suponga una pérdida de generalidad, se supone un  $X_0 = a$  tal que  $\text{signo } f(a) = \text{signo } f''(x) \forall x \in (a, b)$ . La elección de un extremo u otro del intervalo  $(a, b)$  es indiferente, siempre que se cumpla esta condición.

A continuación, se ha de definir  $f(x)$  en términos de su polinomio de Taylor de grado 1, centrado en  $x = a$ , que es sencillamente la recta tangente a la gráfica de la función  $f(x)$  en  $x = a$ .

$$f(x) = f(a) + f'(a) \cdot (x - a) + \frac{f''(c)}{2!} \cdot (x - a)^2 \quad (1)$$

El valor de esta expresión en  $x = r$  es el siguiente:

$$f(r) = 0 = f(a) + f'(a) \cdot (r - a) + \frac{f''(c)}{2!} \cdot (r - a)^2 \quad (2)$$

Si se desarrolla:

$$\begin{aligned} f'(a) \cdot (r - a) &= -f(a) - \frac{f''(c)}{2!} \cdot (r - a)^2 \\ r - a &= -\frac{f(a)}{f'(a)} - \frac{f''(c)}{2f'(a)} \cdot (r - a)^2; f'(a) \neq 0 \\ r &= a - \frac{f(a)}{f'(a)} - \frac{f''(c)}{2f'(a)} \cdot (r - a)^2 = X_1 - \frac{f''(c)}{2f'(a)} \cdot (r - a)^2 \quad (3) \end{aligned}$$

## POLINOMIOS DE TAYLOR

Dada una función  $y = f(x)$ , es posible definir un polinomio *similar* a esta función en las proximidades de un punto  $x = a$ . La definición se realiza en base a las sucesivas derivadas de  $f(x)$ . Cuanto mayor es el grado del polinomio, más parecido es éste a  $f(x)$ . El polinomio de Taylor de grado  $n$  de una función  $f(x)$  centrado en un punto  $x = a$  se define como

$$P_n(x) = f(a) + \frac{f'(a)}{1!} \cdot (x - a) + \frac{f''(a)}{2!} \cdot (x - a)^2 + \dots + \frac{f^{(n)}(a)}{n!} \cdot (x - a)^n$$

Lo que se puede escribir así en forma compacta:

$$P_n(x) = \sum_{n=0}^n \frac{f^{(n)}(a)}{n!} \cdot (x - a)^n$$

De este modo, se puede concluir que

$$f(x) = P_n(x) + \mathcal{R}$$

donde  $\mathcal{R}$  es el resto, un valor que es diferente para cada  $x$  y que, al ser sumado al polinomio de Taylor  $P_n(x)$ , consigue que tenga el mismo *efecto* que  $f(x)$ .

$$\mathcal{R} = \frac{f^{(n+1)}(c)}{(n+1)!} \cdot (x - a)^{n+1}$$

donde  $c$  es un valor entre  $a$  y  $x$  que es diferente para cada  $x$ .

Por la propia definición del método, es un hecho que

$$X_1 = a - \frac{f(a)}{f'(a)}$$

De donde

$$X_1 - a = -\frac{f(a)}{f'(a)}$$

Por lo que

$$\text{signo}(X_1 - a) = \text{signo}\left(-\frac{f(a)}{f'(a)}\right) = -\text{signo}\frac{f(a)}{f'(a)} \quad \mathbf{(4)}$$

De **(3)** se deduce que

$$r - X_1 = -\frac{f''(c)}{2f'(a)} \cdot (r - a)^2$$

Y como  $\text{signo } f''(c) = \text{signo } f'(a)$

$$\text{signo}(r - X_1) = -\text{signo}\frac{f''(c)}{2 \cdot f'(a)} = -\text{signo}\frac{f'(a)}{2 \cdot f'(a)}$$

Evidentemente,

$$\text{signo} \frac{f(a)}{f'(a)} = \text{signo} \frac{f(a)}{2 \cdot f'(a)}$$

Por lo tanto,

$$\text{signo}(X_1 - a) = \text{signo}(r - X_1)$$

$$\begin{cases} X_1 - a > 0 ; r - X_1 > 0 \rightarrow X_1 > a ; r > X_1 \rightarrow \boxed{a < X_1 < r} \\ X_1 - a < 0 ; r - X_1 < 0 \rightarrow X_1 < a ; r < X_1 \rightarrow \boxed{r < X_1 < a} \end{cases}$$

Y, en cualquiera de los dos casos,  $X_1$  estaría comprendido entre  $a$  y  $r$ , lo que asegura la convergencia del método.

## 4. Nuevos métodos planteados.

Ahora se van a presentar los dos métodos desarrollados en el grueso de la investigación, y se va a demostrar su efectividad. En ambos casos, las condiciones iniciales *básicas* son las mismas que las de el método de Newton-Raphson, esto es, se ha de definir una función  $f(x)$ , continua por lo menos en  $[a, b]$ , que tome valores de signo opuesto en  $f(a)$  y en  $f(b)$ , lo que por el teorema de Bolzano significa que existe como mínimo una raíz de la ecuación  $f(x) = 0$  en  $(a, b)$ . Además,  $f''(x)$  ha de tener signo constante en todo  $(a, b)$ . Como condición *adicional*, se exige que  $f'(x)$  también tenga signo constante en  $(a, b)$ .

### 4.1 Método combinado

Este algoritmo consiste en la utilización de dos métodos de forma conjunta. Dado el intervalo  $(a, b)$ , una iteración del método combinado consiste en aplicar el método de Newton-Raphson como se haría con normalidad, es decir, comenzando por el extremo en el que el signo del valor de  $f(x)$  en ese extremo coincide con el signo de  $f''(x)$  en el intervalo de estudio, y aplicar *por el otro extremo* el método de la "regula falsi" (en realidad, se aplica una versión simplificada del mismo, debido a las condiciones adicionales). Esto se hace para conseguir un *intervalo* cada vez más pequeño en el que se sigue encontrando la raíz.

En este punto, vale la pena pararse a ver las diferencias en cuanto a la naturaleza de las soluciones que aportan los métodos dados. El algoritmo de bisección aporta una solución que es un *valor real*, aunque también se puede considerar que da como solución un *intervalo* pues es sencillo pasar de uno a otro en el proceso. Del método de la "regula falsi" se puede decir lo mismo, si bien las operaciones que involucra la conversión de *valor real* a *intervalo* no son tan sencillas. En el caso del método de pendiente fija, y también en el de Newton-Raphson, sin embargo, está claro que la solución que aportan son *valores reales*, en una sucesión. En estos dos métodos sólo se parte de un intervalo inicial para tomar el valor que más convenga; a partir de ese punto los intervalos no se vuelven a ver involucrados en el proceso. En el caso del método combinado, en cada iteración se van calculando dos valores que forman un *intervalo*.

Si se analiza el método de la "regula falsi" con detenimiento, se puede ver que, al no estar entre sus condiciones el hecho de que  $f''(x)$  o  $f'(x)$  tengan signos constantes en  $(a, b)$ , se puede dar el

caso de que las intersecciones de las secantes con el eje  $\overline{OX}$  sean unas mayores y otras menores que  $r$ , esto es, que “caigan” tanto a la izquierda como a la derecha de la raíz. Si esto se diera en el método combinado, se podría llegar a una incongruencia, obteniendo un intervalo que *no* contuviera ninguna raíz en él. Sin embargo, al añadir las dos condiciones citadas arriba, se garantiza que todas las aproximaciones que aporte el método de la “regula falsi” se irán acercando a la raíz por un mismo lado, es decir, a un lado de la raíz se encontrarán las aproximaciones del método de Newton-Raphson, en una sucesión convergente a la raíz y con una monotonía determinada, y al otro lado de la raíz estarán las aproximaciones del método de la “regula falsi”, en sucesión convergente a la raíz pero con la monotonía opuesta, consiguiéndose de este modo intervalos cada vez más pequeños en los que se siga encontrando la raíz.

Concretamente, el método combinado sigue los siguientes pasos:

1. Se considera la función  $y = f(x)$  continua en el intervalo  $[a, b]$  y tal que las imágenes en  $a$  y en  $b$  tienen diferente signo (las imágenes no son nulas ya que, en caso contrario, el problema ya estaría resuelto). Por el Teorema de Bolzano, se sabe que existe al menos un  $r \in (a, b) / f(r) = 0$ . Además, también se ha de verificar que tanto  $f'(x)$  como  $f''(x)$  tengan signo constante en  $[a, b]$ . Si estas condiciones iniciales no son satisfechas, el método finaliza porque no va a funcionar. En otro caso, se sigue con el paso 2.
2. Siendo  $\alpha$  el extremo del intervalo en el que se cumple que  $\text{signo } f(\alpha) = \text{signo } f''(x) \forall x \in (a, b)$ , y  $\beta$  el otro extremo, se halla el valor  $N_1$  siguiendo la sucesión definida por recurrencia que, de hecho, es definitoria del método de Newton-Raphson:

$$\begin{cases} N_0 = \alpha \\ N_{n+1} = N_n - \frac{f(N_n)}{f'(N_n)} \end{cases}$$

3. Se determina la recta  $s_1$ , que pasa por  $P_1 = (\alpha, f(\alpha))$  y por  $Q_1 = (\beta, f(\beta))$ , y corta al eje  $\overline{OX}$  en un punto

$$RF_1 = \frac{f(\beta)\alpha - f(\alpha)\beta}{f(\beta) - f(\alpha)}$$

Nótese que, tal y como ya se ha dicho, el punto  $P_1$  es constante en todas las iteraciones, es decir,  $P_1 = P_2 = P_3 = \dots = P_n = (\alpha, f(\alpha))$ , mientras que  $Q_1$  va cambiando en cada iteración. Así, los siguientes puntos de la iteración por parte de la “regula falsi” vienen dados por la expresión

$$RF_n = \frac{f(RF_{n-1})\alpha - f(\alpha)RF_{n-1}}{f(RF_{n-1}) - f(\alpha)}$$

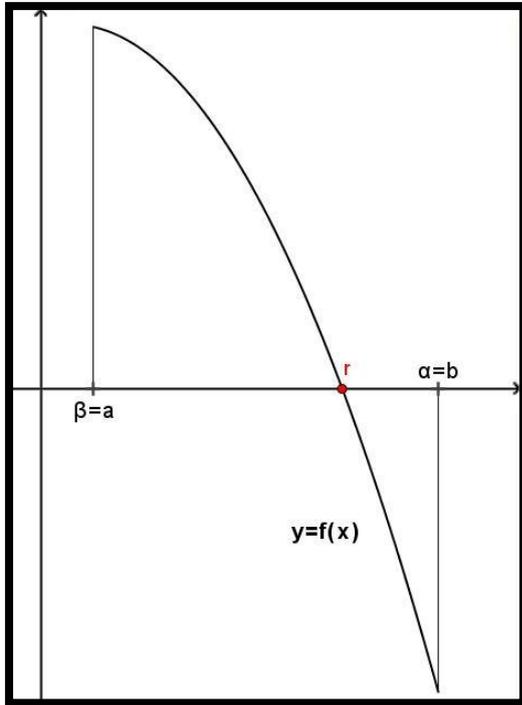
4. Si, en algún caso,  $f(N_n) = 0$  ó  $f(RF_n) = 0$ , la solución se ha encontrado, en caso contrario, se puede continuar repitiendo los pasos 2 y 3 hasta obtener un intervalo de la longitud deseada, o tomar como aproximación un valor (que cumpla la tolerancia requerida) que puede ser: O bien uno de los sucesivos  $N_n$ , o bien uno de los sucesivos  $RF_n$ , o como opción adicional un valor  $c = \frac{N_n + RF_n}{2}$ , o incluso con distinto subíndice (i.e. que pertenezcan a

**MÉTODO COMBINADO. CASOS POSIBLES:** Todas las combinaciones posibles de las condiciones del método configuran estos cuatro casos diferentes. Nótese que el signo de  $f'(x)$  siempre es el mismo que el de  $f(b)$ .

**CASO 1**

$$f(a) > 0 \quad f'(x) < 0$$

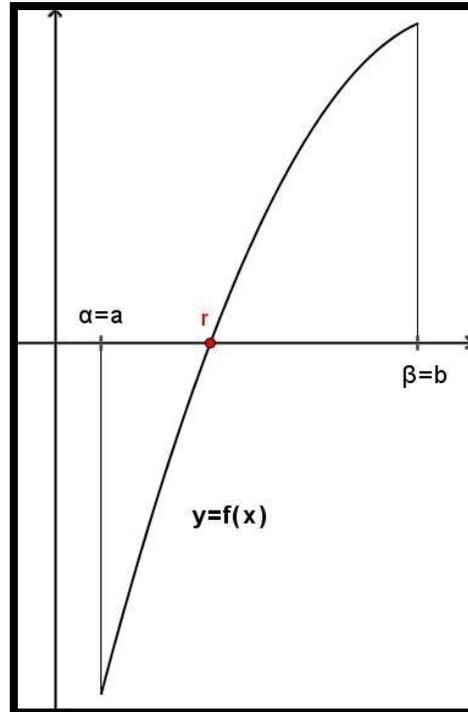
$$f(b) < 0 \quad f''(x) < 0$$



**CASO 3**

$$f(a) < 0 \quad f'(x) > 0$$

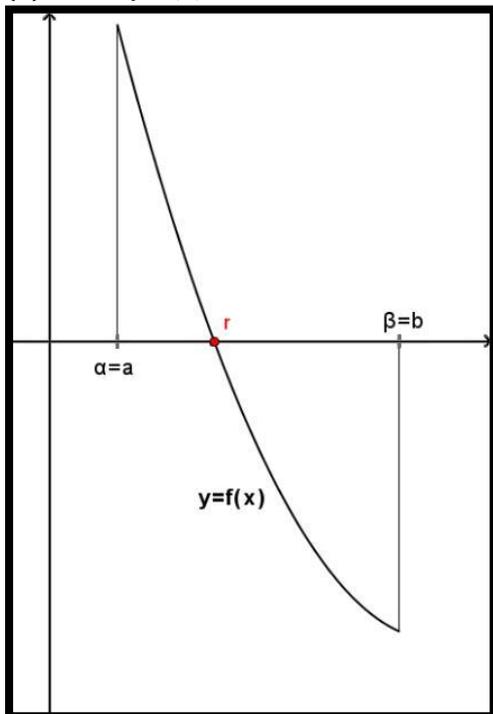
$$f(b) > 0 \quad f''(x) < 0$$



**CASO 2**

$$f(a) > 0 \quad f'(x) < 0$$

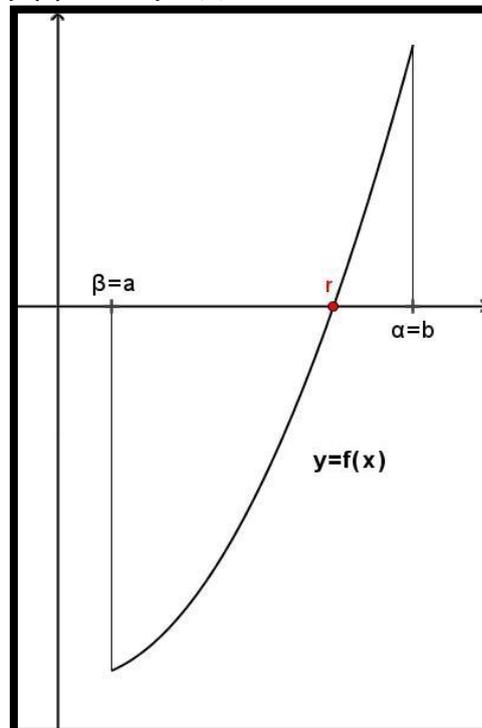
$$f(b) < 0 \quad f''(x) > 0$$



**CASO 4**

$$f(a) < 0 \quad f'(x) > 0$$

$$f(b) > 0 \quad f''(x) > 0$$



distinta iteración). En caso de desear elegir un valor y no un intervalo como resultado final, la elección del mismo se deja en manos del que ejecuta el algoritmo.

**Demostración:**

La demostración de la parte en la que se aplica el método de Newton-Raphson es idéntica a la demostración de este mismo método, expuesta en el apartado 3.4. Sin embargo, queda pendiente demostrar la convergencia del método de “regula falsi” limitado, el que se emplea por el otro extremo. Las condiciones iniciales para esta demostración son:

- Función real de variable real  $y = f(x)$  definida.
- $\exists r \in (a, b) / f(r) = 0 \rightarrow \text{signo } f(a) = -\text{signo } f(b)$
- $\text{signo } f''(x)$  es constante  $\forall x \in (a, b)$
- $\text{signo } f'(x)$  es constante  $\forall x \in (a, b)$

Para facilitar la demostración, se tratarán únicamente los casos 2 y 3. Esta simplificación, si bien descarta dos de los cuatro casos posibles, es suficiente pues los casos eliminados se demostrarían de forma análoga. En los casos que se van a demostrar, se debe probar que  $r < b_1 < b$ .

Se comienza definiendo la recta  $s_1$ , que es la recta secante en la primera iteración. Pasa por los puntos  $A_0 = (a, f(a))$  y  $B_0 = (b, f(b))$ . Por tanto, su vector director es  $\vec{v} = (b - a, f(b) - f(a))$  y, directamente, su pendiente viene dada por la expresión

$$m_{s_1} = \frac{f(b) - f(a)}{b - a}$$

La ecuación de  $s_1$ , definida como la recta de pendiente  $m_{s_1}$  que pasa por  $B$  es la siguiente:

$$s_1 \equiv y - f(b) = \frac{f(b) - f(a)}{b - a} \cdot (x - b)$$

Cuando  $s_1$  corta a  $\overline{OX}$ :

$$0 - f(b) = \frac{f(b) - f(a)}{b - a} \cdot (b_1 - b)$$

Por tanto:

$$b_1 - b = -f(b) \cdot \frac{b - a}{f(b) - f(a)} < 0$$

Esta afirmación se cumple tanto si  $f(a) < 0$  y  $f(b) > 0$  como en el caso contrario. Nótese que  $b - a$  siempre es positivo.

Ahora se trabaja con el desarrollo de Taylor para poner en relación  $b_1$  y  $r$ . Se utilizará el polinomio de Taylor de grado 2, centrado en  $r$ , para su valor en  $b_1$

$$f(b_1) = f(r) + f'(r) \cdot (b_1 - r) + \frac{f''(c)}{2} \cdot (b_1 - r)^2$$

$$f(b_1) = 0 + f'(r) \cdot (b_1 - r) + \frac{f''(c)}{2} \cdot (b_1 - r)^2$$

$$\frac{f(b_1)}{f'(r)} = b_1 - r + \frac{f''(c)}{2f'(r)} \cdot (b_1 - r)^2$$

$$b_1 - r = \frac{f(b_1)}{f'(r)} - \frac{f''(c)}{2f'(r)} \cdot (b_1 - r)^2 > 0$$

De nuevo, el resultado es verdadero en los dos casos analizados.

Las conclusiones obtenidas son las siguientes:

$$\begin{cases} b_1 - b < 0 \\ b_1 - r > 0 \end{cases}$$

Y de ahí:

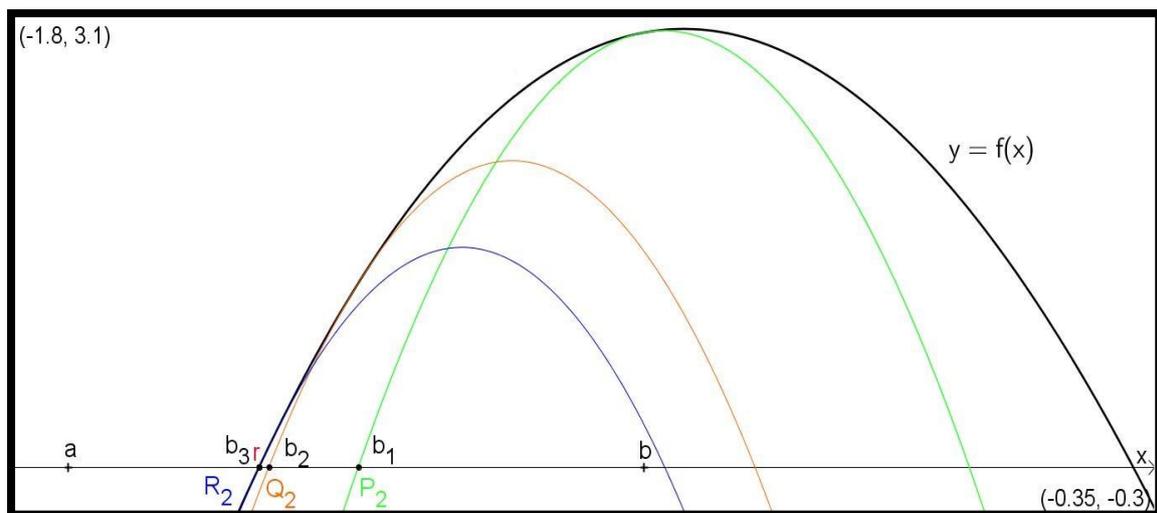
$$b_1 < b; b_1 > r \rightarrow \boxed{r < b_1 < b}$$

## 4.2 Aproximación por parábolas

A continuación se realiza una exposición detallada de todo el proceso por el que se pasó hasta llegar a una conclusión en cuanto a una posible aproximación por parábolas tangentes a la función en estudio.

Al principio se pensó que se podría conseguir algo parecido al método de Newton-Raphson, pero utilizando parábolas tangentes a un punto de la función en vez de rectas tangentes.

Estas parábolas no serían más que el desarrollo de Taylor, centrado en ese punto de la función, y de grado 2, y parecía lógico que el resultado iba a ser algo parecido a la siguiente imagen.



Se conjeturó que la parábola tenía que centrarse en el extremo del intervalo contrario al extremo por el que se empezaría el método de Newton-Raphson, es decir, tenía que centrarse en  $x = \beta$ , y que era necesario que  $f'(x)$  tuviera signo constante en todo  $(a, b)$ . Además, se observó que, de los dos cortes que la parábola tendría con  $\overline{OX}$ , habría que descartar uno de ellos, el que estuviera más alejado de la raíz, siendo el otro la aproximación que interesaría al caso.

A partir de estas ideas iniciales se elaboró una demostración en la que se probaba que el método funcionaría. Según la misma, era posible iterar un proceso en el que se definía la parábola  $P_2(x)$ , centrada en  $x = \beta$ , que cortaría al eje  $\overline{OX}$  en un punto que se encontraría siempre entre  $r$  y  $\beta$ , de modo que las condiciones para la siguiente parábola no variarían, y se podría definir ésta del mismo modo que  $P_2(x)$ , pero centrada en el punto de corte que interesaba. Cada punto de corte, evidentemente, era una nueva aproximación.

Tras realizar esta demostración, entre abril y mayo de 2011, el asunto quedó un tanto apartado, ya que todavía quedaba bastante por hacer en cuanto a todos los demás aspectos del trabajo, y había que terminarlos antes de que se acabara el curso. Por ello, no fue hasta mediados de agosto de 2011 que resultó que la demostración era errónea. Se empezó a sospechar de esto al probar algún cálculo a mano y otros ejemplos en GeoGebra.

El proceso seguido para elaborar esta primera demostración fue similar al de la demostración del método de Newton-Raphson: Se intentó poner en relación el punto de corte,  $X_1$ , con  $\beta$ , y luego estudiar también la relación entre  $X_1$  y  $r$ . Debido a un simple error de despeje, se llegó a unas conclusiones que implicaban que el método convergía. Más específicamente, en la mencionada demostración se probó, erróneamente, que  $X_1$  siempre cortaba a  $\overline{OX}$  entre  $r$  y  $\beta$ , pero al probar varios ejemplos en GeoGebra siempre se obtuvo el resultado contrario:  $X_1$  parecía cortar siempre entre  $\alpha$  y  $r$  (es decir, al otro lado de  $r$ ).

Poco después, se elaboró otro documento, más específico, que trataba los cuatro casos posibles (explicados anteriormente), e intentaba enmendar el error de la primera demostración. En esta segunda prueba, la conclusión que se obtuvo es que no se podía afirmar que el método, con esas condiciones, fuera a converger o no.

Más tarde, se intentó una tercera demostración que pretendía probar que el punto  $X_1$  *siempre* cortaba a  $\overline{OX}$  entre  $\alpha$  y  $r$ , tal y como parecían indicar los resultados en GeoGebra, sin embargo, tampoco se consiguió llegar a una prueba convincente.

De este modo, la situación actual es que la hipótesis planteada en la tercera demostración parece, de momento, ser válida, pero no se ha conseguido demostrar.

Teniendo en cuenta todo esto, se concluyó que el mejor camino que se podía seguir era tener presentes las conclusiones que se han podido sacar a partir del estudio en tres fases que se ha mencionado, e incluirlas como una modificación del método de Newton-Raphson. Hay que remarcar que, al tomar esta decisión, se ha tenido en cuenta que, al ser el interés de la investigación puramente matemático, ha sido clave considerar que las operaciones involucradas son más costosas y de más lenta realización al trabajar con parábolas. Se ha conseguido mantener un cierto equilibrio entre el coste de las operaciones y los beneficios (en términos de rapidez de convergencia) que aporta el uso del método que se va a presentar.

Al igual que en el método combinado, todas las combinaciones posibles de las condiciones del método configuran cuatro casos posibles.

El método modificado se desarrolla, concretamente, de este modo:

1. Se considera la función  $y = f(x)$  continua en el intervalo  $[a, b]$  y tal que las imágenes en  $a$  y en  $b$  tienen diferente signo (las imágenes no son nulas ya que, en caso contrario, el problema ya estaría resuelto). Por el Teorema de Bolzano, se sabe que existe al menos un  $r \in (a, b) / f(r) = 0$ . Además, también se ha de verificar que tanto  $f'(x)$  como  $f''(x)$  tengan signo constante en  $[0, 1]$ . Si estas condiciones iniciales no son satisfechas, el método finaliza porque no va a funcionar. En otro caso, se sigue con el paso 2.
2. Siendo  $\alpha$  el extremo del intervalo en el que se cumple que  $\text{signo } f(\alpha) = \text{signo } f''(x) \forall x \in (a, b)$ , y  $\beta$  el otro extremo, se halla el valor  $N_1$  siguiendo la sucesión definida por recurrencia que, de hecho, es definitoria del método de Newton-Raphson:

$$\begin{cases} N_0 = \alpha \\ N_{n+1} = N_n - \frac{f(N_n)}{f'(N_n)} \end{cases}$$

3. Se determina la parábola  $P_2(x) = f(\beta) + f'(\beta) \cdot (x - \beta) + \frac{f''(\beta)}{2} \cdot (x - \beta)^2$ , que es el desarrollo de Taylor de  $f(x)$  centrado en  $x = \beta$ , y se halla el punto de corte  $PA_1$  que interesa:

$$f(\beta) + f'(\beta) \cdot (x - \beta) + \frac{f''(\beta)}{2} \cdot (x - \beta)^2 = 0$$

$$x - \beta = \frac{-f'(\beta) \pm \sqrt{[f'(\beta)]^2 - 4 \cdot \frac{f''(\beta)}{2} \cdot f(\beta)}}{2 \cdot \frac{f''(\beta)}{2}}$$

$$x = \frac{-f'(\beta) \pm \sqrt{[f'(\beta)]^2 - 4 \cdot \frac{f''(\beta)}{2} \cdot f(\beta)}}{2 \cdot \frac{f''(\beta)}{2}} + \beta$$

Hay que tener en cuenta que si  $\beta = a$  (casos 1 y 4),  $PA_1$  ha de valer el valor mayor (sea  $x_1$  ó  $x_2$ ), y si  $\beta = b$  (casos 2 y 3), el menor.

4. Se comprueba cuál de las dos aproximaciones,  $PA_1$  ó  $N_1$ , es más cercana a  $r$ :
  - a. **Casos 1 y 4:** Si  $PA_1 < N_1$ , entonces  $PA_1$  es mejor aproximación que  $N_1$ , por lo que se continúa con el método de Newton-Raphson, hallando  $N_2$  pero redefiniendo antes  $N_1 = PA_1$ , de modo que  $N_2 = PA_1 - \frac{f(PA_1)}{f'(PA_1)}$ . En caso contrario, se continúa con el método de Newton-Raphson tal y como se había empezado, con  $N_1 = N_1$ .

- b. **Casos 2 y 3:** Si  $PA_1 > N_1$ , entonces  $PA_1$  es mejor aproximación que  $N_1$ , por lo que se continúa con el método de Newton-Raphson, hallando  $N_2$  pero redefiniendo antes  $N_1 = PA_1$ , de modo que  $N_2 = PA_1 - \frac{f(PA_1)}{f'(PA_1)}$ . En caso contrario, se continúa con el método de Newton-Raphson tal y como se había empezado, con  $N_1 = N_1$ .

5. Después de estos cálculos iniciales, el proceso consiste en iterar el método de Newton-Raphson a partir de  $N_1$ , tal y como ya se ha descrito.

De este modo, la ventaja que presenta esta variación del método de Newton-Raphson es que puede llegar a ahorrar algunas iteraciones que habría que realizar si se empleara sólo el método clásico.

## 5. Conclusiones

El resultado principal que se puede obtener del expuesto trabajo es que, si bien el tema elegido como objetivo de la investigación, los algoritmos de aproximación de raíces, ya ha sido estudiado y trabajado en profundidad, todavía se pueden obtener resultados no vistos anteriormente que, aunque de pocas aplicaciones en la vida cotidiana a causa de la venida de fenómenos como la computación, son valiosos porque contribuyen a aumentar nuestro conocimiento en la rama, lo que no es sino un beneficio para el futuro.

Cumpliendo con los objetivos marcados antes incluso de empezar a redactar la presente memoria, se ha hecho un repaso de los métodos más importantes ya existentes, y se ha conseguido entenderlos a fondo. Sin embargo, lo más importante ha sido idear ciertas estrategias que son de utilidad en la aproximación de raíces cuando las circunstancias lo permiten, a partir de las cuales se han desarrollado el método combinado y el modificado. El primero es útil cuando lo que interesa es obtener una mejor acotación de la raíz, mientras que el segundo consigue mejorar la rapidez del método de Newton-Raphson a costa de unas operaciones un tanto más elaboradas. Pese a los problemas acontecidos, se ha conseguido aportar algo que no había sido concebido previamente, lo cual es todo un éxito para la investigación.

## 6. Bibliografía

A lo largo de todo el proceso de elaboración de este trabajo, han sido varios los recursos empleados para la formación, especialmente en lo que concierne al estudio inicial de los algoritmos ya existentes. Las publicaciones más notables empleadas son:

- JAVIER ETAYO; [et al.] 1978. Matemáticas. Madrid, Anaya. Curso de Orientación Universitaria
- FERNANDO CHAMIZO LORENTE, Cálculo Numérico I y II. Universidad Autónoma de Madrid. Problemas resueltos y resúmenes de teoría
- T.M APOSTOL. 1981. Análisis Matemático. Madrid. Reverté.

- R. RIVEROS. 2010. Cálculo Numérico. Pilar. Paraguay. Libros de Cátedra
- MIGUEL ALEMÁN FLORES. Análisis numérico. Universidad de Las Palmas.

Asimismo, los sitios web más relevantes y de mayor utilidad al trabajo son los siguientes:

- <http://www.wiris.net/demo/formula/portal/es/>
- [http://en.wikipedia.org/wiki/Gerolamo\\_Cardano](http://en.wikipedia.org/wiki/Gerolamo_Cardano)
- [http://en.wikipedia.org/wiki/Ars\\_Magna\\_\(Gerolamo\\_Cardano\)](http://en.wikipedia.org/wiki/Ars_Magna_(Gerolamo_Cardano))
- <http://es.wikipedia.org/wiki/Tartaglia>
- [http://en.wikipedia.org/wiki/Lodovico\\_Ferrari](http://en.wikipedia.org/wiki/Lodovico_Ferrari)
- [http://en.wikipedia.org/wiki/Newton%27s\\_method#History](http://en.wikipedia.org/wiki/Newton%27s_method#History)
- [http://en.wikipedia.org/wiki/False\\_position\\_method](http://en.wikipedia.org/wiki/False_position_method)
- [http://neohumanism.org/n/ne/newton\\_s\\_method.html#History](http://neohumanism.org/n/ne/newton_s_method.html#History)
- <http://www.geogebra.org/forum/viewtopic.php?f=20&t=21587>
- <http://www.python.org/>
- <http://code.google.com/p/sympy/>

## ANEXO I: COMPARACIÓN DE LOS ALGORITMOS EXPUESTOS

Comparación de los métodos mostrados en la resolución de la ecuación  $x^3 - 10x^2 + 31x - 14 = 0$  en el intervalo  $(0, 1)$ .

Un valor aproximado de  $r$ , hallado por ordenador (con GeoGebra), y con un redondeo de 15 cifras decimales, es

$$r' = 0.540879841633391$$

Número de iteración	Método de bisección	Método de la "regula falsi"	Método de pendiente fija ( $m = 25$ )	Método de Newton-Raphson	Método combinado	Método modificado
1	0,5	$0, \widehat{63}$	0,56	0,451612903225806 ...	0,451612903225806 ...	0,5361498905772 ...
2	0,75	0,559075907590759 ...	0,54401536	0,537860374121596 ...	0,537860374121596 ...	0,540870965607756 ...
3	0,625	0,544285248381982 ...	0,541377285521735 ...	0,540876221048093 ...	0,540876221048093 ...	0,540879841602052 ...
4	0,5625	0,541514975878	0,540958320586738 ...	0,540879841628177 ...	0,540879841628177 ...	0,540879841633391 ...
5	0,53125	0,540998222732609 ...	0,5408922118039	0,540879841633391 ...	0,540879841633391 ...	-
6	0,546875	-	0,540881791195743 ...	-	-	-

## ANEXO II: EL MÉTODO COMBINADO PROGRAMADO EN PYTHON

A continuación se muestra el código correspondiente al método combinado, programado en Python. Para ejecutarlo, es necesario tener Python instalado, así como el paquete *sympy*.

```
1  #coding=utf-8
2
3  #Importar funciones necesarias
4  from math import sin, cos, tan, asin, acos, atan, sqrt, log, exp
5  from math import sinh, cosh, tanh, asinh, acosh, atanh
6
7  #Importar el paquete sympy (http://sympy.org/) que permitirá la diferenciación de la función de forma automática
8  #(Automatic Differentiation)
9  from sympy import *
10
11 #Se define que x será una variable y por tanto debe ser tratada como un símbolo por sympy
12 x = Symbol ('x')
13
14 #Título
15 print 'MÉTODO COMBINADO programado en Python'
16 print '(Nota: Tenga en cuenta que sólo debe introducir ecuaciones que cumplan con los requisitos del método
17 combinado)'
18 print ' '
19
20 #Petición de datos
21 ecuacion = raw_input('Introduzca la ecuación a resolver en forma de función: f(x)= ')
22 a = float(input('Introduzca el extremo inferior del intervalo en el que se encuentra la raíz: '))
23 b = float(input('Introduzca el extremo superior del intervalo en el que se encuentra la raíz: '))
24 n = int(raw_input('Introduzca el número de iteraciones que desea realizar: '))
25
26 #se hallan las derivadas primera y segunda de la función dada por medio de sympy
27 primera_derivada = str(diff(ecuacion, x))
28 segunda_derivada = str(diff(primer derivada, x))
29
30 #Se definen en Python las funciones a utilizar en el algoritmo
```

```

31 def f(x):
32     return eval(ecuacion)
33 def d(x):
34     return eval(primer derivada)
35 def dd(x):
36     return eval(segunda derivada)
37
38 #Se comprueba cuál es el extremo por el que empezar el método de Newton (alfa) y el otro (beta)
39 if (dd(a) * f(a)) > 0:
40     alfa = a
41     beta = b
42 else:
43     alfa = b
44     beta = a
45
46 #Se definen unas variables necesarias
47 i = 0 #Variable índice para el bucle while
48 N_0 = alfa #Variable que almacena el valor inicial del método de Newton y se actualiza en cada iteración del bucle
49 #con el valor de la aproximación anterior
50 RF = 0 #Variable que almacenará sucesivamente el valor de las iteraciones del método de la "regula falsi". Aquí
51 #sólo se define con el valor 0
52 b_k = b #Variable para el método de la "regula falsi" que se va actualizando según cambia el valor del extremo
53 #que no está fijo (por las condiciones del método, hay uno que está fijo y otro que no lo está)
54
55 #Cuerpo del algoritmo
56 while i < n:
57     N = N_0 - (f(N_0)/d(N_0))
58     N_0 = N
59     i += 1
60     RF = (f(b_k) * a - f(a) * b_k) / (f(b_k) - f(a)) #No es necesario evaluar el signo de RF ya que las condiciones
61     b_k = RF #del problema indican que RF siempre cae entre r y b
62     c = (N + RF) / 2.0
63
64 #Impresión por pantalla de las soluciones obtenidas
65 print ' '
66 print ':: SOLUCIONES APROXIMADAS ::'
67 print ' '

```

```
68 print 'En %d iteraciones del método combinado, los resultados obtenidos en la' % n
69 print 'aproximación de la raíz de %s = 0 en el intervalo (%f, %f) son los siguientes:' % (ecuacion, a, b)
70 print ' '
71 print '·) La última itereación por el lado del método de Newton-Raphson aporta el resultado N =', N
72 print ' '
73 print '·) La última iteración por el lado del método de la "regula falsi" aporta el resultado RF =', RF
74 print ' '
75 print '·) Punto medio final: c =', c
76
77
```