
ALGORITMOS TIPO ILLINOIS PARA LA
RESOLUCIÓN NUMÉRICA DE ECUACIONES NO
LINEALES
Y SU IMPLEMENTACIÓN EN MATLAB[®].

TRABAJO FIN DE GRADO

Autor:

Fernández Palenzuela, Pedro José

Tutor:

Moreno Balcázar, Juan José

GRADO EN MATEMÁTICAS



JUNIO, 2019
Universidad de Almería

Índice general

1	Planteamiento general y objetivos	1
1.1.	Orden de convergencia	1
1.2.	Eficiencia Computacional	2
1.3.	Propiedades de los métodos	3
2	Algunos métodos iterativos de interés	5
2.1.	Métodos de resolución de ecuaciones no lineales	5
	Método Newton, 5.— Método de Bisección, 7.— Método de la Secante, 8.	
2.2.	Método de la Regula Falsi	10
3	Métodos de tipo Illinois	13
3.1.	Método de Illinois	13
3.2.	Método Pegasus	17
3.3.	Métodos de Ford	21
	Método 1, 24.— Método 2, 26.	
	Signo Negativo	27
	Signo Positivo	29
	Método 3, 30.	
4	Implementación en MATLAB[®]	35
4.1.	Código MATLAB [®]	35
	Regula Falsi, 35.— Método de Illinois, 36.— Método de Pegasus, 37.— Métodos de Ford, 38.	
4.2.	Test Numéricos	41
	Funciones del Test, 41.— Resultados y Comparaciones, 43.	
5	Conclusiones	47
	Bibliografía	49
6	Apéndice	I
6.1.	Tablas	I
	Función 2, I.— Función 3, I.— Función 4, I.— Función 5, I.— Función 6, II.— Función 7, II.— Función 8, II.— Función 9, II.— Función de Ford 1, III.— Función de Ford 2, III.	
6.2.	Gráficas	IV
	Función 2, IV.— Función 3, V.— Función 4, VI.— Función 5, VII.— Función 6, VIII.— Función 7, IX.— Función 8, X.— Función 9, XI.— Función de Ford 1, XII.— Función de Ford 2, XIII.	
6.3.	Casos Patológicos	XIV
	Caso 1, XIV.— Caso 2, XIV.— Caso 1, XV.— Caso 2, XVI.	

Abstract in English

In this work we are going to deal with one of the most recurrent problems in numerical analysis, finding the root of a nonlinear equation. We will remember some of the best-known methods to solve this problem where the *Newton's* method stands out but it has a significant inconvenient: it uses the derivative of the function, which means posing an additional condition to the problem that we cannot assure it exists. Even more, it forces the numerical calculation of the evaluations of those derivatives. However, the well-known *Regula Falsi* method uses only the evaluations of the function in some points through an iterative algorithm, with one drawback, the so-called end-point retention. We will develop alternative methods initiated with the papers by Michael Dowell and P. Jarrat giving way to the *Illinois* method and other later generalizations. All these methods solve the end-point retention. We make a study of all of them, obtaining their theoretical efficiencies and later, as a key part of this work, we implement them using MATLAB[®], making a comparison among them and noticing which of them can be more suitable to use. Finally, we establish that these methods are much more effective theoretically and computationally than the *Regula Falsi* method.

Resumen en español

En este trabajo vamos a tratar con uno de los problemas más recurrentes en el análisis numérico: encontrar la raíz de una ecuación no lineal. Recordaremos algunos de los métodos más conocidos para resolver este problema donde destaca el método de Newton pero que posee un inconveniente importante: usa la derivada de la función que se trate siendo un extra que se exige al problema y que no se puede garantizar que se tenga. Además, obliga al cálculo numérico (computacional) de evaluaciones de la derivada. Sin embargo, el conocido *Método de la Regula Falsi* únicamente usa las evaluaciones de la función en determinados puntos a través de un algoritmo iterativo, con un inconveniente: las retenciones de uno de los extremos del intervalo. Desarrollaremos métodos alternativos iniciados con los trabajos de Michael Dowell y P. Jarrat que dan lugar al *Método de Illinois* y a otras generalizaciones posteriores. Todos estos métodos solucionan el problema de la retención de uno de los extremos del intervalo. Hacemos un estudio de ellos, deduciendo su eficiencia teórica y posteriormente, como parte fundamental de este trabajo, implementándolos en MATLAB[®], comparándolos entre ellos y viendo cuales pueden ser más adecuados. Como conclusión estableceremos que estos métodos son mucho más efectivos desde el punto de vista teórico y computacional que el *Método de la Regula Falsi*.

Planteamiento general y objetivos

Un problema clásico en matemáticas es cuando dada una función f no lineal se intentan hallar los valores para los cuales se anula, es decir, los números x tales que

$$f(x) = 0.$$

Este problema ha sido de interés para la comunidad matemática durante mucho tiempo, pues no solo tiene aplicaciones desde el punto de vista teórico sino que una gran cantidad de problemas se pueden expresar como las raíces de una ecuación, incluso otras situaciones de mayor complejidad pueden requerir de la búsqueda de raíces de una función como paso intermedio a encontrar la solución al problema.

Pero la resolución analítica de estas ecuaciones no siempre es posible, usualmente porque no hay método analítico que sea capaz de encontrar una solución. En este contexto es donde aporta el análisis numérico, dando soluciones aproximadas a la real (de hecho, tanto como se desee) mediante algoritmos de diversas características que utilizan como información lo que nos proporciona el problema, desde la función en sí misma, pasando por su evaluación en determinados puntos hasta su derivada de un cierto orden.

Los algoritmos que nosotros vamos a mostrar en este trabajo reciben el nombre de *iterativos* basados en algoritmos que de forma reiterada van obteniendo resultados cada vez más cercanos a la solución buscada. De forma general con x^* la raíz buscada, se plantea generar por iteración una sucesión de elementos x_i , $i = 1, 2, \dots$, tales que

$$\lim_{i \rightarrow \infty} x_i = x^*.$$

Existen varios tipos dentro de los métodos iterativos. Los métodos iterativos de *un paso*, aquellos donde para calcular x_{i+1} solo necesitamos un valor x_i , y los de *n pasos* que para obtener x_{i+1} necesitan los valores de $x_i, x_{i-1}, \dots, x_{i-n+1}$.

También podemos hacer una diferencia entre los métodos *globalmente convergentes* que no tienen restricción alguna a los posibles valores x_i con los que empezar el algoritmo iterativo, y los *localmente convergentes* que no aceptan cualquier conjunto de valores iniciales para llegar a la raíz, estos últimos tienen lo que se llama región de atracción D para la cual cualquier valor $x_i \in D$ permite llegar a la raíz de la función. Normalmente los métodos globales son más lentos que los locales pero a cambio se obtiene la garantía de que cualquier valor permitirá empezar el algoritmo, cosa que con los locales puede requerir de un estudio previo.

Antes de ver cada método en detalle vamos a mostrar un par de conceptos que se encuentran fuertemente relacionados con estos algoritmos.

1.1 Orden de convergencia

Si un cierto algoritmo nos ha dado una sucesión de aproximaciones de la raíz x_1, \dots, x_{i-1}, x_i a raíz de un valor inicial x_0 nos podemos preguntar cual ha sido la velocidad con la que el método ha convergido hasta ese x_i . Podemos hacernos una idea expresando el error de cada paso, es decir, la diferencia absoluta entre la aproximación y el valor de la raíz x^* . De esta forma podemos construir la cadena

$$|x_i - x^*| \approx \rho |x_{i-1} - x^*| \approx \dots \approx \rho^i |x_0 - x^*|.$$

Así cuantificaremos la velocidad de la convergencia en términos de ese ρ . De forma general se suele considerar cuantas iteraciones son necesarias para reducir ese error en un factor fijo de 10, ver [2], es decir, por ejemplo

$$|x_i - x^*| \approx 0,1|x_0 - x^*| \Rightarrow \rho^i \approx 0,1.$$

Tomando logaritmo en base 10 en ambos lados llegamos a $i \log_{10} \rho \approx -1$. A partir de aquí definimos el orden de convergencia como

$$\text{Orden} = -\log_{10} \rho.$$

Una forma de interpretarlo es que hace falta aproximadamente $i = \frac{1}{\text{Orden}}$ iteraciones para reducir el error en el que nos encontremos a un décimo de su valor. Así tenemos que cuanto menor sea ρ mayor el orden de convergencia y con ello menor el número de iteraciones para llegar a la misma reducción del error.

Una definición alternativa que se hace del orden de convergencia consiste en el número ρ que cumple

$$\frac{|x_{i+1} - x^*|}{|x_i - x^*|^\rho} \rightarrow C.$$

Donde C es una constante no cero llamada *constante de error asintótica*.

Pero obtener ese ρ no siempre es fácil, para ninguna de sus dos formas de definirlo. Como se puede deducir para obtener su expresión exacta sería necesario conocer la raíz con la que estamos trabajando lo cual solo es viable en modelos para testear. Aunque el orden de convergencia no es un indicador seguro de la eficiencia de un método, a lo largo de este trabajo habrá métodos cuyo orden de convergencia será mayor que las de otros o menor sin implicar eso que sean métodos más eficientes o menos, respectivamente.

El siguiente concepto, más fiable, es el que usaremos para analizar lo “bueno” de un método.

1.2 Eficiencia Computacional

A la hora de aplicar un método u otro a una cierta ecuación hay poca información que podamos afirmar poseer para cualquier método iterativo que apliquemos. Uno de esos elementos en común son las *unidades de trabajo*, formalmente se definen como el número total d de nuevas evaluaciones de cualquier función que son requeridas por iteración. En [9] nos informan de como Ostrowski en su libro *Konvergenzdiskussion und Fehlerabschätzung für die Newtonsche Methode bei Gleichungssystemen* ha sugerido el nombre de *Horner* como magnitud con la que medir las unidades de trabajo.

A raíz de esta nueva información y haciendo uso del orden de convergencia anterior llegamos a la *Eficiencia Computacional* definida en [9] como

$$\text{Eficiencia} = \rho^{1/d} \tag{1.1}$$

Dando así un indicador del esfuerzo computacional que ejerce un ordenador para resolver el método y llegar a la raíz buscada. Este será el indicador que usemos en este trabajo para señalar cuando un método podría resultar mas conveniente que otro a la hora de compararlos.

Además de estos existen muchos otros conceptos relacionados con los métodos iterativos pero que no consideramos en este trabajo pues o son idénticos o muy similares entre todos los métodos que vamos a estudiar o no aportan nada a nuestro caso.

1.3 Propiedades de los métodos

- **Precisión:** una magnitud muy importante a la hora de diseñar métodos numéricos. Establece la magnitud del error que se espera cuando realizamos un método. En nuestro caso al estar tratando con métodos iterativos la precisión queda en un segundo plano puesto que para aumentar la precisión del resultado que obtenemos bastaría con seguir haciendo nuevas iteraciones.
- **Robustez:** una magnitud más unida al software que usemos para resolver un método que con su realización en sí. Permite asegurar que un método vaya a funcionar bajo cualquier condición, esto es, bajo niveles de errores de tolerancia aceptables que no vayan a desencadenar en un error del software. Ninguno de los métodos que usaremos tendrá problemas de robustez.
- **Condicionamiento y estabilidad:** el primero, más relacionado con el problema que tratemos en sí, nos dice que un problema se llama *mal condicionado* si una pequeña perturbación de los datos iniciales del mismo desencadena en un resultado final muy diferente, dificultando encontrar un algoritmo para resolverlo que cumpla nuestras condiciones de robustez. Y un algoritmo es *estable* si este no aumenta los errores de redondeo que cometemos cuando resolvemos el problema. Juntando ambas definiciones, si un problema está *bien condicionado* (es decir, no está mal condicionado) y es estable entonces el resultado que obtenemos computacionalmente está muy cerca del resultado exacto del problema.

El objetivo de este trabajo es mostrar una serie de métodos iterativos nacidos a raíz de la *Regula Falsi* y compararlos para ver cual es el más recomendable de usar de todos ellos. Una vez descritos todos estos conceptos podemos introducir las secciones de nuestro trabajo. En el siguiente capítulo vamos a analizar una serie de métodos para la búsqueda de raíces de una ecuación, donde destacaremos la *Regula Falsi* y un inconveniente en las funciones sobre intervalos cóncavos y convexos. A continuación analizaremos una serie de nuevos métodos que son modificaciones de la *Regula Falsi* para solventar ese problema, surgidos a raíz del llamado *Método de Illinois*, observando su estructura y eficiencia computacional. Terminando el trabajo con la implementación de todos ellos en MATLAB[®] pudiendo comparar su comportamiento con una serie de funciones de elaboración propia y otras de los artículos originales de los nuevos métodos.

Algunos métodos iterativos de interés

2.1 Métodos de resolución de ecuaciones no lineales

Dentro del análisis numérico podemos encontrar varios métodos para la resolución del problema

$$f(x) = 0, \quad (2.1)$$

cuando f no es lineal. Vamos a analizar los más conocidos de ellos y estudiar el por qué son menos preferibles ante la *Regula Falsi*.

Método Newton

El método más conocido y estudiado para resolver ecuaciones, es el *Método de Newton* (o de *Newton-Raphson*).

La mecánica del método se basa en suponer que tenemos una función f continua y derivable en un cierto intervalo y tratamos de resolver (2.1). Si tenemos que x^* es la raíz y x es una aproximación podemos hacer la suposición de que $x^* = x + h$ para un cierto h . De esta forma llegamos a una aproximación lineal a través del desarrollo de Taylor

$$0 = f(x^*) \simeq f(x) + (x^* - x)f'(x) = f(x) + hf'(x) \rightarrow h \simeq -\frac{f(x)}{f'(x)},$$

de forma que sustituyendo h en la relación que hemos supuesto entre la raíz real y la aproximación tenemos

$$x^* \simeq x - \frac{f(x)}{f'(x)}.$$

Ahora podemos obtener de él un método iterativo dado por esta última fórmula

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i \in \mathbb{N}, \quad (2.2)$$

donde partimos de un valor inicial x_0 , tratándose así de un método de un solo paso. Entonces repetimos con el nuevo punto la operación y así sucesivamente hasta cumplir un cierto criterio de parada que suele ser que el valor absoluto de la diferencia de dos pasos consecutivos sea menor que un cierto valor, es decir, detendremos el método si $|x_{i+1} - x_i| < \epsilon$ para un cierto ϵ que nosotros elijamos, normalmente llamado **tolerancia**.

Geoméricamente se puede ver que x_{k+1} es la intersección de la tangente de la gráfica de f en el punto $(x_k, f(x_k))$ con el eje OX .

Pero a partir de aquí empiezan a surgir los problemas con el *Método de Newton*. Para empezar sólo podemos asegurar una convergencia en las proximidades de la raíz, es decir posee una convergencia local, lo cual no es de extrañar, el método se ha derivado del desarrollo de Taylor de la función en la raíz despreciando a partir de la segunda derivada inclusive. Por lo tanto, es posible que estemos incurriendo en un error muy grande a menos que estemos cogiendo puntos muy cercanos a la raíz de forma que la parte eliminada del truncamiento del desarrollo de Taylor sea despreciable.

Si quisiéramos ampliar esta convergencia de forma que pudiéramos coger un punto cualquiera y llegar a la raíz, es decir una convergencia global, es necesario cumplir una

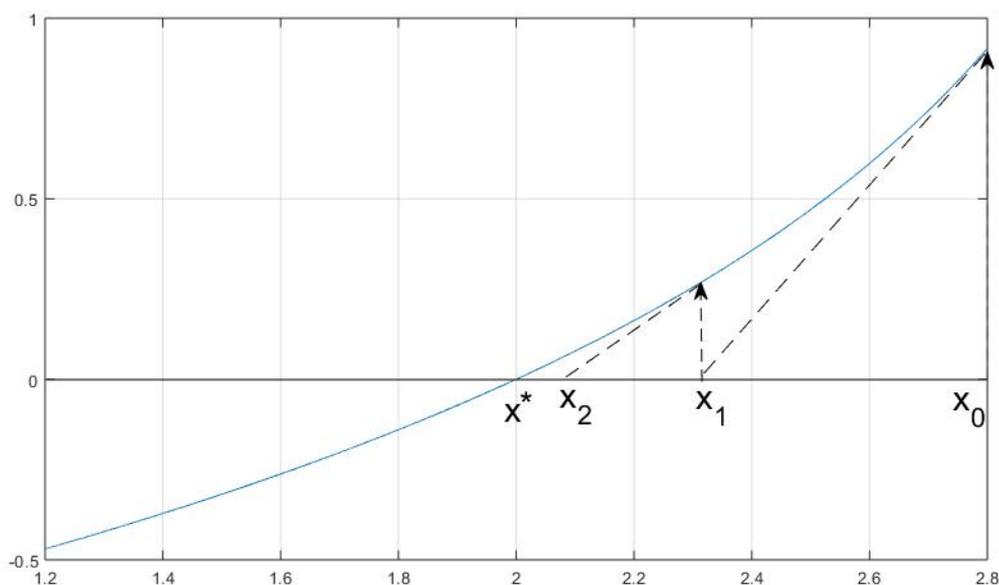


Figura 2.1: Método de Newton

serie de condiciones resumidas en dos teoremas que nos aporta [8] y que vemos a continuación.

Teorema 2.1. Si $f \in C^2(\mathbb{R})$ es estrictamente creciente y convexa en todo \mathbb{R} , y tiene un cero en \mathbb{R} , entonces el cero es único y la iteración de Newton converge a él a partir de cualquier valor inicial x_0 .

Teorema 2.2. Sea $f \in C^2([a, b])$. Si $f(a) \cdot f(b) < 0$, y $f'(x)$ y $f''(x)$ son no nulas y conservan el signo en $[a, b]$, entonces, a partir de la aproximación inicial $x_0 \in [a, b]$ satisfaciendo $f(x_0) \cdot f''(x_0) > 0$, el método de Newton converge hacia la raíz única de $f(x)$ en $[a, b]$.

Como podemos apreciar en un vistazo rápido son condiciones exigentes y restrictivas. El Teorema 2.1 pide continuidad y derivabilidad en todo \mathbb{R} lo cual para un polinomio no hay problema pero en general no tiene por qué ser cierto y más la convexidad y crecimiento estricto. Como vemos este primer teorema apenas nos da libertad a la hora de utilizar el método, por ello lo normal es recurrir al segundo teorema que se restringe a un intervalo, pero este tampoco está libre de exigencias. Como estamos ante funciones que hemos supuesto continuas y buscamos una raíz de estas, la condición de $f(a) \cdot f(b) < 0$ no presenta gran problema pero de nuevo nos presenta la necesidad de la existencia de hasta una doble derivada pero aquí se exige que éstas no sean nulas y conserven el signo en el intervalo encontrado, esta condición básicamente nos obliga a buscar un intervalo sumamente pequeño en la mayoría de las situaciones no triviales. De esta forma la convergencia se vuelve local, algo a tener muy en cuenta puesto que no siempre vamos a poder reducir el intervalo todo lo que deseemos o no tendremos los recursos para ello, pero como beneficio en [2] nos demuestran como el método tiene un orden de convergencia cuadrático, es decir $\rho = 2$.

Veamos un método donde las exigencias sobre f son menores y la convergencia es global.

Método de Bisección

Es considerado el método más sencillo de todos puesto que únicamente se apoya en un resultado.

Teorema 2.3 (Teorema de Bolzano). *Sea f una función real continua en el intervalo cerrado $[a, b]$ con las evaluaciones de los extremos de distinto signo, $f(a) \cdot f(b) < 0$. Entonces existe al menos un punto $c \in (a, b)$ tal que $f(c) = 0$*

La idea del *Método de la Bisección* es usar reiteradamente el *Teorema de Bolzano*. Se escoge un intervalo que cumpla la sencilla condición descrita antes y se parte el intervalo en dos y nos quedamos con el intervalo que siga cumpliendo con la condición. Esto claramente no nos aporta una velocidad de convergencia muy buena puesto que por ejemplo si la raíz se encuentra muy cerca de uno de los extremos del intervalo el método no hace nada por acortar el camino, estará haciendo subdivisiones conservando en todas uno de los extremos fijos por lo que si además los dos extremos del intervalo inicial estaban muy alejados se puede requerir de muchas iteraciones.

Formalmente partimos de un intervalo $[a, b]$ y se van a construir una serie de intervalos uno dentro del anterior. Si consideramos $a_0 = a$ y $b_0 = b$, partiendo del intervalo genérico $[a_i, b_i]$ construimos el punto

$$x_i = \frac{a_i + b_i}{2}, \quad (2.3)$$

y hacemos la comprobación

- si $f(a_i) \cdot f(x_i) < 0$ entonces $a_{i+1} = a_i$ y $b_{i+1} = x_i$, teniendo nuevo intervalo $[a_{i+1}, b_{i+1}]$,
- si $f(b_i) \cdot f(x_i) < 0$ entonces $a_{i+1} = x_i$ y $b_{i+1} = b_i$, teniendo nuevo intervalo $[a_{i+1}, b_{i+1}]$.

Y volvemos a repetir el procedimiento con el nuevo intervalo hasta establecer un criterio de parada de entre las múltiples variantes que existen. Todas ellas dependerán de un cierto ϵ que establecemos antes como una tolerancia:

- i) $|b_{i+1} - a_{i+1}| < \epsilon$, es decir, que el tamaño del intervalo sea menor que la tolerancia. Quedándonos con cualquiera de los dos extremos como nuestra raíz, o incluso con el punto medio entre los dos.
- ii) $|f(x_i)| < \epsilon$, es decir, que la evaluación del último punto obtenido sea menor que la tolerancia. Quedándonos con este último punto calculado como la raíz.
- iii) Que el número de iteraciones sea igual a κ , es decir, ponemos un limitador al número de veces que se puede repetir el método. Quedándonos con el último valor calculado antes de alcanzar el límite de iteraciones.

(2.4)

Detallemos un poco más el criterio de parada i) que será el que usaremos en este método. Podemos saber cual será la longitud del intervalo i -ésimo usando (2.3), así

$$|b_i - a_i| = \frac{|b_0 - a_0|}{2^i}. \quad (2.5)$$

Veámoslo, por ejemplo si suponemos que los extremos de cada intervalo de la iteración que permanecen fijos se van intercaldando, viendo que se haría de forma análoga para cualquier intercambio de extremos,

$$\begin{aligned} |b_i - a_i| &= \left| b_{i-1} - \frac{a_{i-1} + b_{i-1}}{2} \right| = \left| \frac{b_{i-1} - a_{i-1}}{2} \right| = \frac{|b_{i-1} - a_{i-1}|}{2} = \\ &= \frac{\left| \frac{a_{i-2} + b_{i-2}}{2} - a_{i-2} \right|}{2} = \frac{\left| \frac{b_{i-2} - a_{i-2}}{2} \right|}{2} = \frac{|b_{i-2} - a_{i-2}|}{2^2} = \dots \end{aligned}$$

y de esa forma podemos ver como llegamos a (2.5).

De esta manera podríamos llegar a saber cuantas iteraciones del método necesitamos para alcanzar cierta tolerancia, simplemente usando (2.5) y suponiendo que llegamos a cumplir la condición en la iteración i -ésima,

$$|b_i - a_i| = \frac{|b_0 - a_0|}{2^i} < \epsilon \Leftrightarrow i > \frac{\ln\left(\frac{|b-a|}{\epsilon}\right)}{\ln 2},$$

luego se efectuarán tantas iteraciones como el menor número entero natural que cumpla la desigualdad anterior.

De manera similar a lo antes desarrollado podemos demostrar que se trata de un método de convergencia global, no importa cuan grande sera el intervalo inicial que cojamos, siempre se llegará a una solución si es raíz simple. Si suponemos que x^* es la raíz, por la forma que hemos construido el método y haciendo uso del *Teorema de Bolzano*, la raíz siempre se encontrará en el intervalo, luego si estamos en la i -ésima iteración,

$$|x^* - x_i| = \frac{|b_0 - a_0|}{2^i} \Rightarrow \lim_{i \rightarrow \infty} |x^* - x_i| = 0 \Leftrightarrow \lim_{i \rightarrow \infty} x_i = x^*.$$

Tenemos su convergencia global, pero esto no quiere decir que sea rápida. También necesitamos matizar que se trata de un método para encontrar raíces simples dentro del intervalo, en el caso en el que haya raíces múltiples dentro del intervalo el proceso no garantiza que converja a una raíz.

Podemos apreciar ciertas ventajas respecto al *Método de Newton* como puede ser su fácil realización puesto que sólo necesitamos continuidad de la función en el intervalo que estemos trabajando sin ser necesaria ninguna condición de derivabilidad y evaluar el signo de la misma en ciertos puntos aunque tiene un grave inconveniente y es su falta de velocidad. Su convergencia está asegurada pero a un precio de una lentitud exagerada en cuanto el intervalo inicial sea grande, algo que puede surgir a menudo, de hecho en [2] nos muestran como no es posible afirmar que el *Método de Bisección* tenga un orden convergencia lineal, es decir $\rho < 1$.

Método de la Secante

Para este método suponemos que tenemos dos pares de puntos $(x_{i-1}, f(x_{i-1}))$ y $(x_i, f(x_i))$ que son aproximaciones de la raíz y construimos la recta que los une. Esta deberá de cortar el eje de abscisas en un cierto punto x_{i+1} consiguiendo un nuevo punto $(x_{i+1}, f(x_{i+1}))$. Por tanto, si utilizamos la pendiente de la recta que hemos creado con

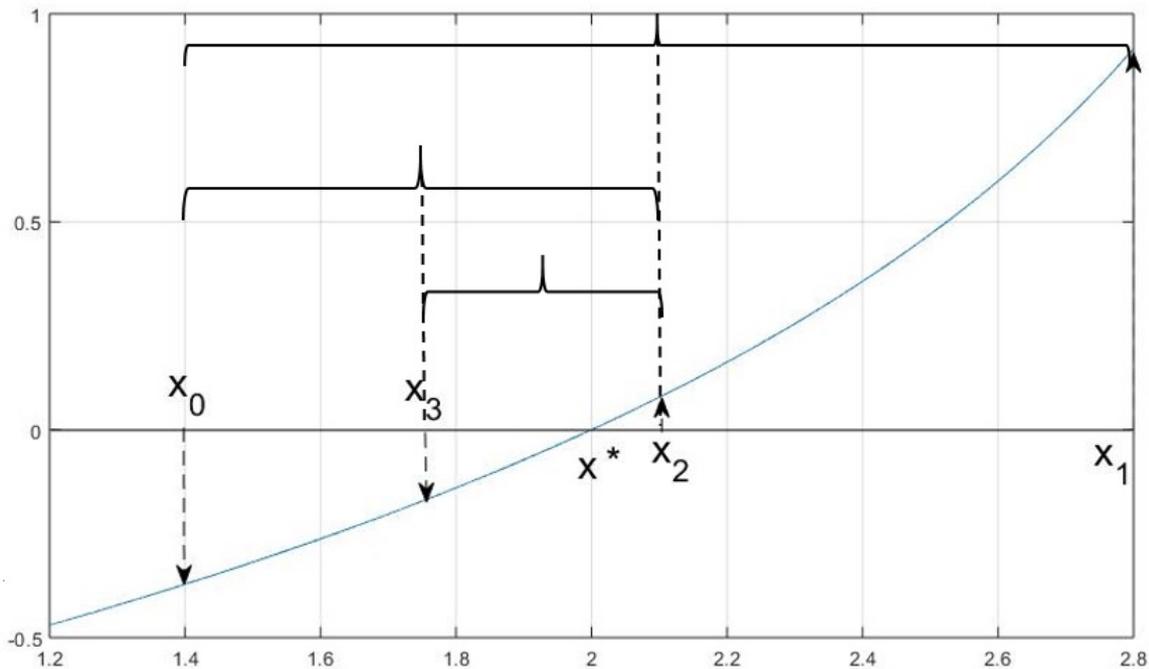


Figura 2.2: Método de Bisección

los puntos \$(x_{i-1}, f(x_{i-1}))\$ y \$(x_i, f(x_i))\$ y despejando el punto de corte \$x_{i+1}\$ obtenemos la fórmula

$$x_{i+1} = x_i - \frac{(x_i - x_{i-1})f(x_i)}{f(x_i) - f(x_{i-1})} \quad (2.6)$$

A diferencia del *Método de Bisección* el *Método de la Secante* no hace ninguna comprobación adicional para comprobar cuales serán los extremos del intervalo de la siguiente iteración, esto es debido a que el *Método de la Secante* no hace uso del *Teorema de Bolzano 2.3* por lo tanto nunca estamos pretendiendo en este método dar un intervalo donde se encontrará la raíz. Esto da lugar a que no sea necesario que los dos puntos iniciales que usemos cumplan que \$f(x_i)f(x_{i-1}) < 0\$ provocando que gráficamente el comportamiento del algoritmo pueda llegar a parecer un tanto caótico en entornos que no sean exclusivamente cóncavos o convexos. Una vez calculado el nuevo punto \$x_{i+1}\$ para calcular el \$x_{i+2}\$ solo hay que atenerse a la fórmula, que usará \$x_{i+1}\$ y \$x_i\$. De esta manera no se usará el mismo punto más de dos veces en este método.

Aunque fueron desarrollados de forma independiente, el *Método de la Secante* se puede obtener de la expresión (2.2) que usábamos en el *Método de Newton* usando la aproximación numérica de orden 1 de la primera derivada

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}},$$

quedándonos con la misma fórmula

$$x_{i+1} = x_i - \frac{f(x_i)}{\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}} = x_i - \frac{(x_i - x_{i-1})f(x_i)}{f(x_i) - f(x_{i-1})}.$$

Respecto a su convergencia no podemos afirmar que sea global, de hecho para unos datos iniciales x_0 y x_1 muy lejanos entre sí el método no asegura convergencia. En cambio tenemos el siguiente teorema para la convergencia local, su demostración se puede ver en [8] páginas 99-101.

Teorema 2.4. *Sea $f \in C(I)$, donde I es un entorno adecuado de la raíz x^* de f , y si $f''(x^*) \neq 0$. Entonces para todo par de valores iniciales x_0 y x_1 en I lo suficientemente cerca de x^* , el método (2.6) converge a x^* con orden $\rho = \frac{1+\sqrt{5}}{2} \approx 1,618$.*

Geoméricamente ya hemos visto a la hora de describir el método que consiste en crear la recta que une dos puntos x_{i-1} y x_i y obtener la evaluación de la función en el punto donde esta recta interseque al eje de abscisas.

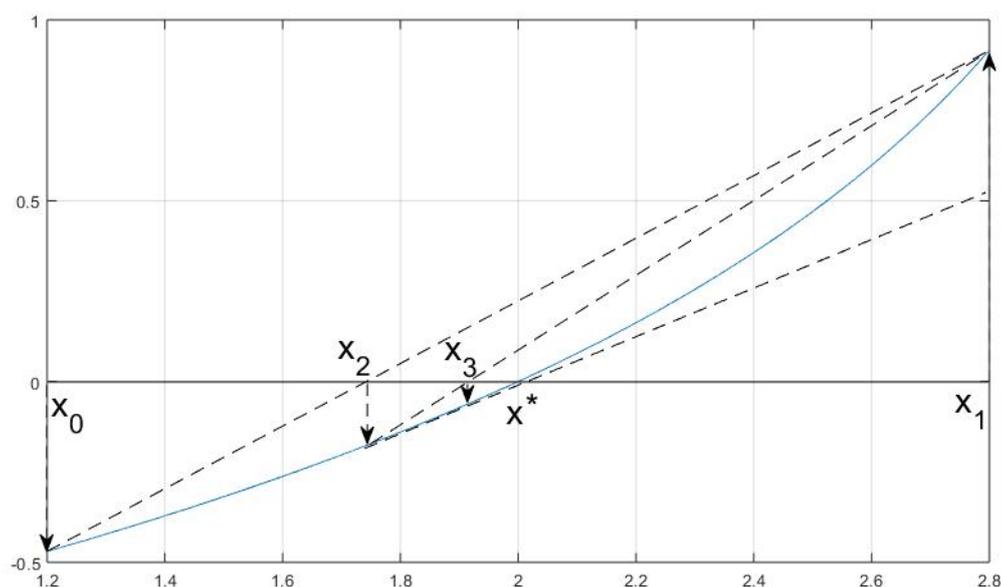


Figura 2.3: Método de la Secante

Comparándolo con los métodos anteriores hemos vuelto a salvar el problema que teníamos con el *Método de Newton* con la primera derivada f' y no ser necesario el uso del *Teorema de Bolzano* 2.3, aunque estas ventajas son a cambio de no asegurar una convergencia global, únicamente local.

El siguiente método, denominado de la *Regula Falsi*, intenta solventar las dificultades encontradas en los dos métodos descritos anteriormente, mezclando la estructura algorítmica del método de la secante con el uso del *Teorema de Bolzano*.

2.2 Método de la Regula Falsi

El método de la falsa posición o regla falsi es un método clásico y bien estudiado cuyas modificaciones serán el motivo de este trabajo. A continuación se describe el método de la *Regula Falsi* original.

Partiendo de un intervalo $[x_{i-1}, x_i]$ que cumpla la condición del *Teorema de Bolzano*, calculamos la siguiente aproximación de la raíz como sigue

$$x_{i+1} = x_i - \frac{(x_i - x_{i-1})f(x_i)}{f(x_i) - f(x_{i-1})} \quad (2.7)$$

$$\begin{aligned} &= x_{i-1} - \frac{(x_i - x_{i-1})f(x_{i-1})}{f(x_i) - f(x_{i-1})} \\ &= \frac{x_i f(x_{i-1}) - x_{i-1} f(x_i)}{f(x_{i-1}) - f(x_i)}. \end{aligned} \quad (2.8)$$

De esta forma generamos un nuevo intervalo que dependerá de la evaluación del nuevo punto x_{i+1} ,

- si $f(x_{i-1}) \cdot f(x_{i+1}) < 0$ entonces tenemos el nuevo intervalo $[x_{i-1}, x_{i+1}]$,
- si $f(x_i) \cdot f(x_{i+1}) < 0$ entonces tenemos el nuevo intervalo $[x_{i+1}, x_i]$.

Así siempre tenemos a la raíz que buscamos dentro del intervalo por el *Teorema de Bolzano* y, al igual que pasaba en el *Método de la Bisección*, nuestra convergencia está asegurada de esta manera. Al igual que antes podemos elegir diversos criterios de parada tales como la diferencia de ambos extremos como en los métodos anteriores $|x_i - x_{i-1}| < \epsilon$ o que la evaluación en valor absoluto del último punto calculado sea menor que la tolerancia prefijada, $|f(x_{i+1})| < \epsilon$ de forma que ese punto que evaluamos será nuestra aproximación de la raíz x^* .

Geométricamente estamos usando la recta que une los dos extremos del intervalo para luego coger la evaluación del punto resultante del cruce de esa recta con el eje OX y volver a repetir el procedimiento.

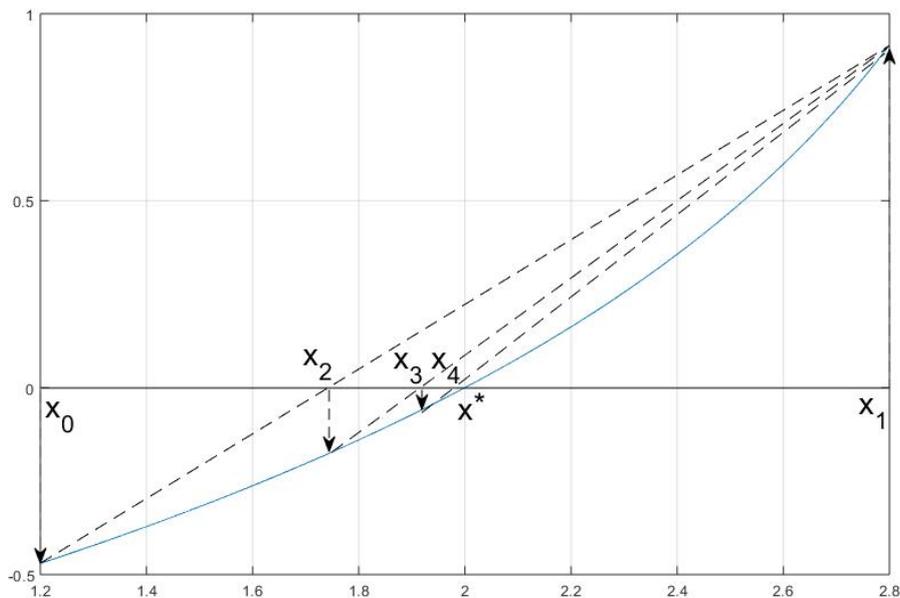


Figura 2.4: Método de la Regula Falsi

Por lo tanto volvemos a tener un método sencillo que no nos exige apenas condiciones iniciales mas allá de las del *Teorema de Bolzano* 2.3. Sin embargo la velocidad de convergencia se ve reducida a nivel lineal siempre que la función f en el intervalo sea cóncava o convexa debido a que uno de los extremos del intervalo permanece fijo durante todo el proceso, se convierte en un punto muerto, lo cual ralentiza mucho el método.

El motivo de esta retención es muy sencillo, como tratamos con funciones cóncavas o convexas toda recta que una dos puntos del dominio quedará por debajo o por encima de nuestra función, respectivamente. Por lo tanto si llamamos $g(x)$ a la recta que une dos puntos cualesquiera, nuestra función $f(x)$ cumplirá que $g(x) < f(x)$ ó $g(x) > f(x)$ según sea cóncava o convexa. Luego el punto de corte $g(x) = 0$ nos indica que $f(x) > 0$ o $f(x) < 0$ respectivamente, generando esta relación siempre en todo el dominio que escojamos. Así solamente se podrá mantener la condición del Teorema de Bolzano si el extremo de signo contrario a la $f(x)$ antes descrita permanece fijo. Este motivo hizo que la *Regula Falsi* quedase algo más en segundo plano comparada con otros métodos iterativos como el visto *Método de Newton* aunque no tenga los problemas de convergencia que tiene éste. Porque no debemos de olvidar que la *Regula Falsi* siempre converge a raíces reales simples a nivel global, su único inconveniente es que su velocidad de convergencia asintótica pasa a ser de orden 1 cuando nos encontramos con el problema de la concavidad y la convexidad.

En este trabajo vamos a presentar unas modificaciones a la *Regula Falsi* que se han ido haciendo a lo largo del tiempo empezando en 1971 con el llamado *Método de Illinois* para solventar este problema que posee el método, convirtiéndolo así en un método mucho más atractivo de usar dado su sencillez. Después veremos el *Método de Pegasus* de los mismo autores que el *Método de Illinois* que mejora los resultados de este para finalizar con tres métodos que fueron presentados en el mismo artículo por un nuevo autor.

Métodos de tipo Illinois

En 1971 Michael Dowel y P. Jarrat en [3] presentan una modificación del clásico método de resolución de ecuaciones no lineales denominado *Regula Falsi* y descrito en el capítulo anterior. En dicho trabajo los autores afirman que la idea que van a desarrollar fue usada originalmente por los empleados del centro de informática de la Universidad de Illinois alrededor de 1950, de ahí el nombre que recibirá la modificación, *Método de Illinois*. Son M. Dowel y P. Jarrat quienes hacen un análisis más exhaustivo del método y de sus mejoras respecto de la *Regula Falsi* estándar.

A lo largo de los siguientes años, guiados por el espíritu del *Método Illinois*, fueron apareciendo nuevos métodos que también modificaban la *Regula Falsi*, mejorando su rendimiento y evitando sus problemas, siendo incluso un adelanto respecto al *Método de Illinois*. Se trata de dos métodos que recibieron un estudio de forma individual y luego otros tres que fueron trabajados como conjunto en el mismo artículo. Nosotros los vamos a tratar todos, empezando por el que da nombre a este TFG, el *Método de Illinois*.

3.1 Método de Illinois

La intención del método es solucionar el problema de los extremos fijos que puede ocurrir en la *Regula Falsi* estándar. Este nuevo método sigue el mismo procedimiento de la *Regula Falsi*. Veamos como calcular el nuevo punto x_{i+1} de forma que la raíz buscada $x^* \in (x_i, x_{i+1})$. En primer lugar usamos la *Regula Falsi* (2.8), es decir,

$$x_{i+1} = \frac{x_i f_{i-1} - x_{i-1} f_i}{f_{i-1} - f_i},$$

donde hemos usado la notación

$$f_i = f(x_i), \quad \forall i,$$

para facilitar la lectura del trabajo.
A continuación

- i) si $f_i \cdot f_{i+1} < 0$ entonces (x_{i-1}, f_{i-1}) se reemplaza por (x_i, f_i) y $x^* \in (x_i, x_{i+1})$,
- ii) si $f_i \cdot f_{i+1} > 0$ entonces (x_{i-1}, f_{i-1}) se reemplaza por $(x_{i-1}, f_{i-1}/2)$ y $x^* \in (x_{i-1}, x_{i+1})$.

De esta manera estamos diciendo que cuando estamos en la situación ii), vamos a calcular el siguiente punto de manera similar pero atendiendo a una fórmula ligeramente modificada

$$x_{i+2} = \frac{x_{i+1} \frac{f_{i-1}}{2} - x_{i-1} f_{i+1}}{\frac{f_{i-1}}{2} - f_{i+1}}. \quad (3.1)$$

Este cálculo del nuevo punto evita las retenciones por un extremo fijo que ocurre en la *Regula Falsi*. En la Figura 3.1 podemos observar gráficamente como la retención en los extremos se evita. Es decir, geoméricamente es tan sencillo como que para un paso modificado la pendiente de la recta que existía en la *Regula Falsi* ahora tendrá la mitad de la pendiente.

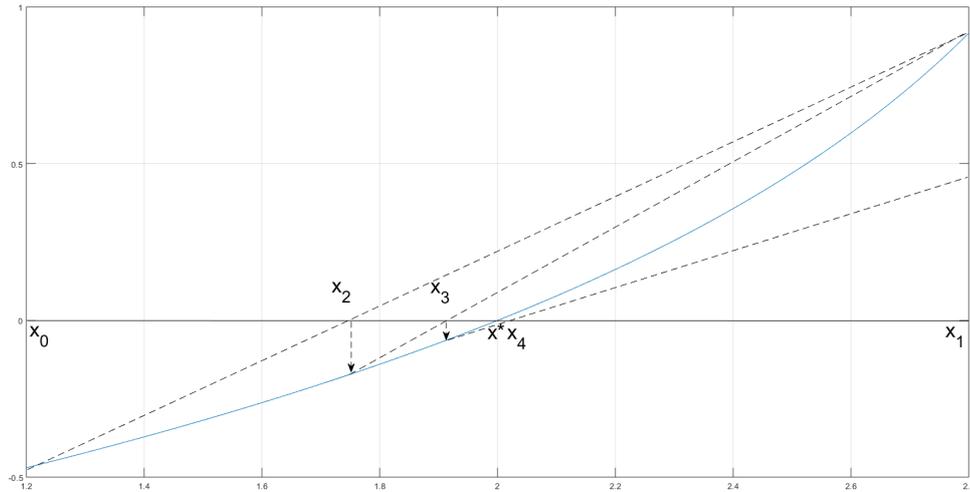


Figura 3.1: Método de Illinois

Pasemos ahora a analizar la convergencia del método, recordemos que la *Regula Falsi* es de orden 1 debido a su problema con la concavidad y convexidad. Debemos de encontrar una mejora a esta velocidad para que todo este nuevo método sea de utilidad real. Si partimos de unas estimaciones x_0 y x_1 que hemos encontrado lo suficientemente cerca de la raíz x^* de (2.1), podemos usar teoría de errores para ver con precisión como se comporta el algoritmo que tratamos. Comenzamos definiendo un error para la aproximación i -ésima

$$\epsilon_i = x_i - x^*, \quad (3.2)$$

y haciendo uso del desarrollo de Taylor para nuestra función f en la raíz x^* y evaluando en x_i llegamos

$$f_i = \sum_{k=1}^{\infty} c_k \epsilon_i^k, \quad (3.3)$$

donde

$$c_i = \frac{f^{(i)}(x^*)}{i!} \quad \text{y} \quad c_0 = f(x^*) = 0. \quad (3.4)$$

Usando (2.7) podemos tener una expresión aproximada del error para la primera iteración

$$x_{i+1} = x_i - \frac{(x_i - x_{i-1})f_i}{f_i - f_{i-1}} \Leftrightarrow x_{i+1} - x_i = -\frac{(x_i - x_{i-1})f_i}{f_i - f_{i-1}}.$$

Ahora se cumple que

$$x_{i+1} - x_i = x_{i+1} - x^* - x_i + x^* = \epsilon_{i+1} - \epsilon_i.$$

Luego

$$\begin{aligned} \epsilon_{i+1} - \epsilon_i = -\frac{(x_i - x_{i-1})f_i}{f_i - f_{i-1}} \Leftrightarrow \epsilon_{i+1} &= \epsilon_i - \frac{(x_i - x^* - x_{i-1} + x^*)f_i}{f_i - f_{i-1}} = \frac{\epsilon_i f_i - \epsilon_i f_{i-1} - f_i \epsilon_i + f_i \epsilon_{i-1}}{f_i - f_{i-1}} \\ &= \frac{-\epsilon_i f_{i-1} + f_i \epsilon_{i-1}}{f_i - f_{i-1}}. \end{aligned}$$

Usando ahora el desarrollo de Taylor hasta la segunda derivada (no olvidemos que $f(x^*) = 0$ por ser la raíz)

$$\begin{aligned} \frac{-\epsilon_i f_{i-1} + f_i \epsilon_{i-1}}{f_i - f_{i-1}} &\sim \frac{-\epsilon_i \left[f'(x^*) \epsilon_{i-1} + \frac{f''(x^*)}{2} \epsilon_{i-1}^2 \right] + \epsilon_{i-1} \left[f'(x^*) \epsilon_i + \frac{f''(x^*)}{2} \epsilon_i^2 \right]}{f'(x^*) \epsilon_i - f'(x^*) \epsilon_{i-1}} \\ &= \frac{-\epsilon_i \frac{f''(x^*)}{2} \epsilon_{i-1}^2 + \epsilon_{i-1} \frac{f''(x^*)}{2} \epsilon_i^2}{f'(x^*) (\epsilon_i - \epsilon_{i-1})} = \frac{\epsilon_i \epsilon_{i-1} f''(x^*) (\epsilon_i - \epsilon_{i-1})}{2 f'(x^*) (\epsilon_i - \epsilon_{i-1})} = \frac{f''(x^*)}{2 f'(x^*)} \epsilon_i \epsilon_{i-1}. \end{aligned}$$

Así hemos llegado a que el error de la primera iteración es aproximadamente

$$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1} \right) \epsilon_i \epsilon_{i-1}. \quad (3.5)$$

Para la siguiente iteración tenemos una bifurcación pues la expresión dependerá de si vamos a aplicar o no una iteración modificada, es decir, depende de si $f_i f_{i+1}$ es menor o mayor que cero.

Para un paso sin modificar, i.e. $f_i f_{i+1} < 0$, aplicamos (3.5) de forma directa y obtenemos

$$\epsilon_{i+2} \sim \left(\frac{c_2}{c_1} \right) \epsilon_{i+1} \epsilon_i,$$

pero para un paso modificado el estudio del error nos revela que si consideramos un intervalo lo suficientemente pequeño tenemos, usando (3.1)

$$\epsilon_{i+2} = x_{i+2} - x^* = \frac{f_{i-1} \epsilon_{i+1} - 2 f_{i+1} \epsilon_{i-1}}{f_{i-1} - 2 f_{i+1}},$$

volvemos a hacer el desarrollo de Taylor como en la anterior demostración y obtenemos, usando la notación (3.4)

$$\frac{(c_1 \epsilon_{i-1} + c_2 \epsilon_{i-1}^2) \epsilon_{i+1} - 2(c_1 \epsilon_{i+1} + c_2 \epsilon_{i+1}^2) \epsilon_{i-1}}{c_1 \epsilon_{i-1} + c_2 \epsilon_{i-1}^2 - 2c_1 \epsilon_{i+1} - 2c_2 \epsilon_{i+1}^2} = \frac{\epsilon_{i-1}^2 \epsilon_{i+1} - c_2 \epsilon_{i+1}^2 \epsilon_{i-1} - c_1 \epsilon_{i+1} \epsilon_{i-1}}{c_1 \epsilon_{i-1} + c_2 \epsilon_{i-1}^2 - 2c_1 \epsilon_{i+1} - 2c_2 \epsilon_{i+1}^2}.$$

aquí es donde tendremos en cuenta que hacemos un estudio de su comportamiento asintótico quedándonos así con los elementos más grandes de cada parte del cociente que por como funciona el método será el ϵ de subíndice más pequeño ϵ_{i-1} , y como por hipótesis hemos supuesto que el intervalo es suficientemente pequeño podemos afirmar que $|\epsilon_{i-1}| < 1$. Así nos quedamos con que

$$\epsilon_{i+2} \sim \frac{-c_1 \epsilon_{i+1} \epsilon_{i-1}}{c_1 \epsilon_{i-1}} = -\epsilon_{i+1}. \quad (3.6)$$

Ahora usando tanto (3.5) como (3.6) podemos estudiar el comportamiento asintótico del *Método de Illinois* que podremos apreciar que funciona por patrones. Si suponemos que ϵ_{i-1} es negativo y ϵ_i positivo (cosa coherente puesto que $x^* \in [x_{i-1}, x_i]$ y por definición $\epsilon_{i-1} = x_{i-1} - x^*$ y $\epsilon_i = x_i - x^*$) junto con $\frac{c_2}{c_1}$ también positivo podremos observar una secuencia de valores obtenidos usando un paso modificado (M) o uno no modificado (U) de forma UUM, UUM,... Veamos que esto es así, consideremos de forma concreta que $c_1 > 0$ y $c_2 > 0$ teniendo así una función convexa creciente, luego si $\epsilon_i < 0$ implica que $f_i < 0$ y $\epsilon_i > 0$ implica que $f_i > 0$, $\forall i$ y recordando (3.2), (3.5) y (3.6), tenemos

- $$\left. \begin{aligned} \epsilon_{i-1} < 0 &\Rightarrow x_{i-1} < x^* \Rightarrow f_{i-1} < 0 \\ \epsilon_i > 0 &\Rightarrow x_i > x^* \Rightarrow f_i > 0 \end{aligned} \right\}$$

luego $f_{i-1}f_i < 0$, tenemos un paso no modificado U.

- $$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1}\right)\epsilon_i\epsilon_{i-1} \Rightarrow \epsilon_{i+1} < 0 \Rightarrow x_{i+1} < x^* \Rightarrow f_{i+1} < 0,$$

luego $f_i f_{i+1} < 0$, tenemos un paso no modificado U.

- $$\epsilon_{i+2} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+1}\epsilon_i \Rightarrow \epsilon_{i+2} < 0 \Rightarrow x_{i+2} < x^* \Rightarrow f_{i+2} < 0,$$

luego $f_{i+1}f_{i+2} > 0$, tenemos un paso modificado M.

- $$\epsilon_{i+3} \sim -\epsilon_{i+2} \Rightarrow \epsilon_{i+3} > 0 \Rightarrow x_{i+3} > x^* \Rightarrow f_{i+3} > 0,$$

luego $f_{i+2}f_{i+3} < 0$, tenemos un paso no modificado U.

- $$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+3}\epsilon_{i+2} \Rightarrow \epsilon_{i+4} < 0 \Rightarrow x_{i+4} < x^* \Rightarrow f_{i+4} < 0,$$

luego $f_{i+3}f_{i+4} < 0$, tenemos un paso no modificado U.

- $$\epsilon_{i+5} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+4}\epsilon_{i+3} \Rightarrow \epsilon_{i+5} < 0 \Rightarrow x_{i+5} < x^* \Rightarrow f_{i+5} < 0,$$

luego $f_{i+4}f_{i+5} > 0$, tenemos un paso modificado M.

Y así podemos apreciar el patrón UUM antes mencionado. Se haría una demostración análoga para el caso donde $c_1 < 0$ y $c_2 < 0$ junto con $\epsilon_{i-1} > 0$ y $\epsilon_i < 0$.

Para $\frac{c_2}{c_1}$ negativo llegamos a una secuencia similar pero que comienza diferente UM, UUM, UUM, \dots , pudiendo observar rápidamente que el patrón base acaba siendo dos pasos sin modificar seguido de uno modificado. Veamos que esto es así, en este caso consideraremos $c_1 > 0$ pero $c_2 < 0$ de forma que estamos ante una función cóncava creciente por lo tanto al igual que antes si $\epsilon_i < 0$ implica que $f_i < 0$ y $\epsilon_i > 0$ implica que $f_i > 0$, $\forall i$

- $$\left. \begin{aligned} \epsilon_{i-1} < 0 &\Rightarrow x_{i-1} < x^* \Rightarrow f_{i-1} < 0 \\ \epsilon_i > 0 &\Rightarrow x_i > x^* \Rightarrow f_i > 0 \end{aligned} \right\}$$

luego $f_{i-1}f_i < 0$, tenemos un paso no modificado U.

- $$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1}\right)\epsilon_i\epsilon_{i-1} \Rightarrow \epsilon_{i+1} > 0 \Rightarrow x_{i+1} > x^* \Rightarrow f_{i+1} > 0,$$

luego $f_i f_{i+1} > 0$, tenemos un paso modificado M.

■

$$\epsilon_{i+2} \sim -\epsilon_{i+1} \Rightarrow \epsilon_{i+2} < 0 \Rightarrow x_{i+2} < x^* \Rightarrow f_{i+2} < 0,$$

luego $f_{i+1}f_{i+2} < 0$, tenemos un paso no modificado U.

■

$$\epsilon_{i+3} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+2}\epsilon_{i+1} \Rightarrow \epsilon_{i+3} > 0 \Rightarrow x_{i+3} > x^* \Rightarrow f_{i+3} > 0,$$

luego $f_{i+2}f_{i+3} < 0$, tenemos un paso no modificado U.

■

$$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+3}\epsilon_{i+2} \Rightarrow \epsilon_{i+4} > 0 \Rightarrow x_{i+4} > x^* \Rightarrow f_{i+4} > 0,$$

luego $f_{i+3}f_{i+4} > 0$, tenemos un paso modificado M.

Y así podemos apreciar el inicio con *UM* seguido del patrón *UUM* antes mencionado. Se haría una demostración análoga para el caso donde $c_1 < 0$ y $c_2 > 0$ junto con $\epsilon_{i-1} > 0$ y $\epsilon_i < 0$.

Teniendo así, usando de nuevo (3.5) y (3.6), que estos patrones tienen un error de la forma

$$\epsilon_{i+2} \sim \left(\frac{c_2}{c_1}\right)^2 \epsilon_{i-1}^3,$$

de esta forma que si el error de cada patrón lo unimos como si se tratara de una sola iteración, llamémoslo μ_i , tenemos

$$\mu_{i+1} \sim \left(\frac{c_2}{c_1}\right)^2 \mu_i^3.$$

De esta forma estamos diciendo que nos encontramos ante un método que es de tercer orden al coste de 3 evaluaciones de f por paso. Si usamos la definición de eficiencia computacional de [9, Apéndice C] tenemos que es $3^{\frac{1}{3}} = 1,442\dots$, que se trata de una eficiencia mayor a la de 1 que tenía la *Regula Falsi* estándar que trabajamos en (2.8). De esta forma, a cambio de una modificación mínima, que en esencia se ha basado en dividir entre dos por lo que el coste computacional es insignificante, hemos obtenido una mejora de la eficiencia computacional de casi del 45%.

Una vez hemos visto como funciona el *Método de Illinois* vamos a empezar a trabajar con el resto de métodos, estos funcionan de manera idéntica, buscan encontrar un valor γ que, multiplicado a f_{i-1} , optimice la velocidad de convergencia, de la forma

$$x_{i+1} = \frac{x_i(\gamma f_{i-1}) - x_{i-1}f_i}{(\gamma f_{i-1}) - f_i}.$$

Empecemos a estudiar los dos métodos que han recibido un estudio individual.

3.2 Método Pegasus

Un año después de su *Método de Illinois*, en 1972, M. Dowell y P. Jarrat presentaron una nueva modificación de la *Regula Falsi* usando el mismo espíritu que el anterior,

aunque esta vez el valor γ no es constante, es adaptativo a cada paso que nos encontremos.

Así el esquema tras calcular un primer punto queda

- i) si $f_i \cdot f_{i+1} < 0$ entonces (x_{i-1}, f_{i-1}) se reemplaza por (x_i, f_i) y $x^* \in (x_i, x_{i+1})$,
- ii) si $f_i \cdot f_{i+1} > 0$ entonces (x_{i-1}, f_{i-1}) se reemplaza por $(x_{i-1}, f_{i-1}f_i/(f_{i+1} + f_i))$ y $x^* \in (x_{i-1}, x_{i+1})$.

Como habíamos dicho con el *Método de Illinois* en este caso recalculamos el punto x_{i+1} con la fórmula algo más compleja

$$x_{i+1} = \frac{x_i \frac{f_{i-1}f_i}{f_{i+1}+f_i} - x_{i-1}f_i}{\frac{f_{i-1}f_i}{f_{i+1}+f_i} - f_i}. \quad (3.7)$$

Geoméricamente estamos reduciendo f_{i-1} mediante el factor $\frac{f_i}{f_{i+1}+f_i}$ para prevenir los extremos fijos que tanto nos reducían la velocidad de convergencia en la *Regula Falsi*. Lo volvemos a observar gráficamente.

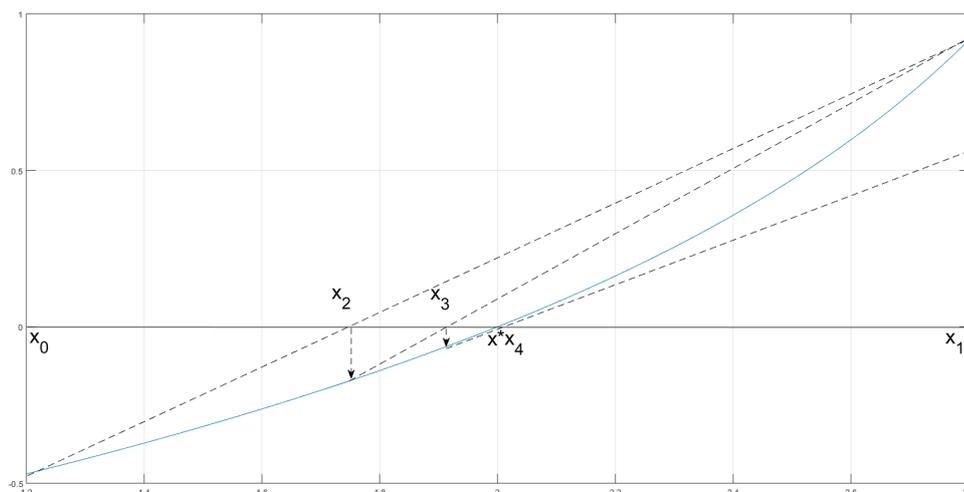


Figura 3.2: Método de Pegasus

Para el análisis de la convergencia volvemos a usar todo lo que habíamos definido en el *Método de Illinois*, es decir, la expresión (3.2) para ϵ_i , el desarrollo de Taylor (3.3) y la representación de c_i (3.4).

Las expresiones para los errores de los pasos sin modificar sigue siendo la misma, (3.5), como no podía ser de otra manera, ahora para un paso modificado en el cual $f_i f_{i+1}$ es positivo, se obtiene

$$\epsilon_{i+2} \sim \left(\frac{c_2}{c_1} \right)^2 \epsilon_{i+1} \epsilon_{i-1}^2. \quad (3.8)$$

En [4] los autores prueban que

$$\epsilon_{i+2} = c \epsilon_{i+1}^{d_1},$$

donde d_1 es la raíz real del polinomio $x^3 - x^2 - 2 = 0$ y c una constante cualquiera. Por tanto $d_1 = 1,69562\dots$

Observando (3.8) nos podemos dar cuenta rápidamente que asintóticamente ϵ_{i+1} y ϵ_{i+2} tendrán el mismo signo, por lo tanto se volverá a tener el mismo signo a la hora de obtener $f_i f_{i+1}$, luego un primer paso modificado (M_1) siempre será seguido de un segundo paso modificado (M_2). Si avanzamos una iteración más podemos llegar a que

$$\epsilon_{i+3} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+2} \epsilon_{i+1},$$

volviendo así de nuevo a tener la expresión sin modificar (3.5).

De la misma forma que con el *Método de Illinois* ahora podemos estudiar el comportamiento asintótico una vez tenemos todos estos errores calculados. Asumimos nuevamente que ϵ_{i-1} es negativo y que ϵ_i es positivo y empezando con $\frac{c_2}{c_1}$ positivo llegamos a la secuencia de valores expresados en función de si fueron modificados (M_1 y M_2) o no (U) de la forma $UUM_1M_2, UUM_1M_2, \dots$. Veamos al igual que con el *Método de Illinois* que esto es así, consideremos de forma concreta que $c_1 > 0$ y $c_2 > 0$ teniendo así una función convexa creciente, luego si $x_i < x^*$ implica que $f_i < 0$ y $x_i > x^*$ implica que $f_i > 0$, $\forall i$ y con la expresión de los errores antes descritos a la vez que recordando (3.2) y (3.5)

$$\left. \begin{array}{l} \epsilon_{i-1} < 0 \Rightarrow x_{i-1} < x^* \Rightarrow f_{i-1} < 0 \\ \epsilon_i > 0 \Rightarrow x_i > x^* \Rightarrow f_i > 0 \end{array} \right\}$$

luego $f_{i-1} f_i < 0$, tenemos un paso no modificado U.

$$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1}\right) \epsilon_i \epsilon_{i-1} \Rightarrow \epsilon_{i+1} < 0 \Rightarrow x_{i+1} < x^* \Rightarrow f_{i+1} < 0,$$

luego $f_i f_{i+1} < 0$, tenemos un paso no modificado U.

$$\epsilon_{i+2} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+1} \epsilon_i \Rightarrow \epsilon_{i+2} < 0 \Rightarrow x_{i+2} < x^* \Rightarrow f_{i+2} < 0,$$

luego $f_{i+1} f_{i+2} > 0$, tenemos un paso modificado M_1 .

$$\epsilon_{i+3} \sim \left(\frac{c_2}{c_1}\right)^2 \epsilon_{i+2} \epsilon_i^2 \Rightarrow \epsilon_{i+3} < 0 \Rightarrow x_{i+3} < x^* \Rightarrow f_{i+3} < 0,$$

luego $f_{i+2} f_{i+3} > 0$, tenemos un paso modificado M_2 .

$$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+3} \epsilon_{i+2} \Rightarrow \epsilon_{i+4} > 0 \Rightarrow x_{i+4} > x^* \Rightarrow f_{i+4} > 0,$$

luego $f_{i+3} f_{i+4} < 0$, tenemos un paso no modificado U.

$$\epsilon_{i+5} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+4} \epsilon_{i+3} \Rightarrow \epsilon_{i+5} < 0 \Rightarrow x_{i+5} < x^* \Rightarrow f_{i+5} < 0,$$

luego $f_{i+4} f_{i+5} < 0$, tenemos un paso no modificado U.

▪

$$\epsilon_{i+6} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+5} \epsilon_{i+4} \Rightarrow \epsilon_{i+6} < 0 \Rightarrow x_{i+6} < x^* \Rightarrow f_{i+6} < 0,$$

luego $f_{i+5}f_{i+6} > 0$, tenemos un paso modificado M_1 .

▪

$$\epsilon_{i+7} \sim \left(\frac{c_2}{c_1}\right)^2 \epsilon_{i+6} \epsilon_{i+4}^2 \Rightarrow \epsilon_{i+7} < 0 \Rightarrow x_{i+7} < x^* \Rightarrow f_{i+7} < 0,$$

luego $f_{i+6}f_{i+7} > 0$, tenemos un paso modificado M_2 .

Y así podemos apreciar el patrón UUM_1M_2 antes mencionado. Una demostración análoga se puede hacer para el caso donde $c_1 < 0$ y $c_2 < 0$ junto con $\epsilon_{i-1} > 0$ y $\epsilon_i < 0$.

Para el caso de $\frac{c_2}{c_1}$ negativo el comportamiento es similar, si en el *Método de Illinois* ocurría que antes de seguir su patrón aparecía un paso sin modificar U seguido de otro modificado M, aquí ocurrirá lo mismo pero aparecen los dos pasos sin modificar M_1 y M_2 . Es decir tenemos un patrón UM_1M_2, UUM_1M_2, \dots . Veamos que esto es así, en este caso consideraremos $c_1 > 0$ pero $c_2 < 0$ de forma que estamos ante una función cóncava creciente por lo tanto al igual que antes si $x_i < x^*$ implica que $f_i < 0$ y $x_i > x^*$ implica que $f_i > 0, \forall i$.

▪

$$\left. \begin{array}{l} \epsilon_{i-1} < 0 \Rightarrow x_{i-1} < x^* \Rightarrow f_{i-1} < 0 \\ \epsilon_i > 0 \Rightarrow x_i > x^* \Rightarrow f_i > 0 \end{array} \right\}$$

luego $f_{i-1}f_i < 0$, tenemos un paso no modificado U.

▪

$$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1}\right) \epsilon_i \epsilon_{i-1} \Rightarrow \epsilon_{i+1} > 0 \Rightarrow x_{i+1} > x^* \Rightarrow f_{i+1} > 0,$$

luego $f_i f_{i+1} > 0$, tenemos un paso modificado M_1 .

▪

$$\epsilon_{i+2} \sim \left(\frac{c_2}{c_1}\right)^2 \epsilon_{i+1} \epsilon_{i-1}^2 \Rightarrow \epsilon_{i+2} > 0 \Rightarrow x_{i+2} > x^* \Rightarrow f_{i+2} > 0,$$

luego $f_{i+1}f_{i+2} > 0$, tenemos un paso modificado M_2 .

▪

$$\epsilon_{i+3} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+2} \epsilon_{i+1} \Rightarrow \epsilon_{i+3} < 0 \Rightarrow x_{i+3} < x^* \Rightarrow f_{i+3} < 0,$$

luego $f_{i+2}f_{i+3} < 0$, tenemos un paso no modificado U.

▪

$$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+3} \epsilon_{i+2} \Rightarrow \epsilon_{i+4} > 0 \Rightarrow x_{i+4} > x^* \Rightarrow f_{i+4} > 0,$$

luego $f_{i+3}f_{i+4} < 0$, tenemos un paso no modificado U.

▪

$$\epsilon_{i+5} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+4} \epsilon_{i+3} \Rightarrow \epsilon_{i+5} > 0 \Rightarrow x_{i+5} > x^* \Rightarrow f_{i+5} > 0,$$

luego $f_{i+4}f_{i+5} > 0$, tenemos un paso modificado M_1 .

$$\epsilon_{i+6} \sim \left(\frac{c_2}{c_1}\right)^2 \epsilon_{i+5} \epsilon_{i+3}^2 \Rightarrow \epsilon_{i+6} > 0 \Rightarrow x_{i+6} > x^* \Rightarrow f_{i+6} > 0,$$

luego $f_{i+5}f_{i+6} > 0$, tenemos un paso modificado M_2 .

Esto genera un patrón completo de 4 subpasos y una relación final dada en [4] de la forma

$$\epsilon_{i+8} \sim \left(\frac{c_2}{c_1}\right)^8 \epsilon_i^2 \epsilon_{i+4}^7,$$

concluyendo en un proceso de orden 7.275 al coste de 4 evaluaciones de la función f . Recurriendo otra vez a la eficiencia computacional de [9, Apéndice C] tenemos $7,275^{1/4} = 1,642\dots$, de nuevo mayor que la *Regula Falsi* con una mejora de casi el 65% que a su vez supera a la del *Método de Illinois* alrededor de un 15%. Por lo tanto nos encontramos con una nueva modificación de la *Regula Falsi* que aumenta significativamente la eficiencia computacional conservando las buenas cualidades de la *Regula Falsi*, sin requerir cambios drásticos, simplemente evaluaciones de la función que ya teníamos calculados.

3.3 Métodos de Ford

Hemos llamado *Métodos de Ford* a aquellos métodos que proporcionan nuevos valores del parámetro γ que definimos anteriormente. Estudiemos dichos métodos.

De manera ideal querríamos tener un valor de γ el cual haga que la secante que une los puntos $(x_{i-1}, \gamma f_{i-1})$ y (x_{i+1}, f_{i+1}) pasase por la raíz x^* que estamos buscando. En [5] obtienen la relación

$$\frac{\gamma f_{i-1}}{x_{i-1} - x^*} = \frac{f_{i+1}}{x_{i+1} - x^*},$$

de donde

$$\gamma = \frac{f_{i+1}}{x_{i+1} - x^*} \frac{x_{i-1} - x^*}{f_{i-1}}. \quad (3.9)$$

Evidentemente esto no es computable, buscamos precisamente el valor de x^* , pero podemos aproximar partes de esa expresión por otras que conocemos obteniendo una nueva fórmula que dará lugar a nuevos, y en algunos casos, mejores métodos. Como x^* es la raíz de f ($f(x^*) = 0$), escribimos (3.9) de la forma

$$\begin{aligned} \gamma &= \frac{\frac{f_{i+1} - f(x^*)}{x_{i+1} - x^*}}{\frac{f_{i-1} - f(x^*)}{x_{i-1} - x^*}} \\ &= \frac{f[x_{i+1}, x^*]}{f[x_{i-1}, x^*]}, \end{aligned} \quad (3.10)$$

donde hemos introducido la notación de diferencia dividida

$$f[x_i, x_j] = \frac{f_i - f_j}{x_i - x_j}.$$

Este sería el valor idóneo para γ . Haciendo uso de las pendientes que unen $(x_{i-1}, \gamma f_{i-1})$ y (x_{i+1}, f_{i+1}) con $(x^*, f(x^*))$. El estudio ahora radica en encontrar una posible aproximación general para x^* que vayamos a poder usar en (3.10).

En general, ¿cómo podemos aproximar x^* en la expresión (3.10)? Ford utiliza la derivada del polinomio cuadrático interpolador de f en tres posibles puntos x_i, x_{i-1} y x_{i+1} . Aquí [9] nos aporta una expresión para la derivada de la interpolación cuadrática en x_i , aunque en este caso será para una permutación $\{j, k, l\}$ cualquiera de $\{i-1, i+1, i\}$, dando lugar a la expresión

$$f[x_l, x_k] + f[x_l, x_j] - f[x_k, x_j].$$

No resulta difícil obtener dicha expresión a raíz del polinomio interpolador de los tres puntos

$$f(x_l) + f[x_l, x_k](x - x_l) + f[x_l, x_k, x_j](x - x_l)(x - x_k)$$

si ahora derivamos la expresión tenemos

$$f[x_l, x_k] + f[x_l, x_k, x_j](2x - (x_k + x_l))$$

si ahora lo evaluamos en x_l tenemos

$$\begin{aligned} f[x_l, x_k] + f[x_l, x_k, x_j](x_l - x_k) &= f[x_l, x_k] + (x_l - x_k) \frac{f[x_l, x_k] - f[x_k, x_j]}{x_l - x_j} \\ &= f[x_l, x_k] + (x_l - x_k) \frac{\frac{f(x_l) - f(x_k)}{x_l - x_k} - \frac{f(x_k) - f(x_j)}{x_k - x_j}}{x_l - x_j} \\ &= f[x_l, x_k] + (x_l - x_k) \frac{x_k f(x_l) - x_k f(x_j) - x_j f(x_l) + x_j f(x_k) - x_l f(x_k) + x_l f(x_j)}{(x_k - x_j)(x_l - x_j)(x_l - x_k)} \\ &= f[x_l, x_k] + \frac{x_k f(x_l) - x_k f(x_j) - x_j f(x_l) + x_j f(x_k) - x_l f(x_k) + x_l f(x_j)}{(x_k - x_j)(x_l - x_j)(x_l - x_k)} \end{aligned}$$

ahora sumamos y restamos $x_j f(x_j)$ en el numerador del cociente y podemos agrupar

$$\begin{aligned} f[x_l, x_k] + \frac{[x_k f(x_l) - x_k f(x_j) - x_j f(x_l) + x_j f(x_j)] - [x_l f(x_k) - x_l f(x_j) - x_j f(x_k) + x_j f(x_j)]}{(x_k - x_j)(x_l - x_j)} \\ = f[x_l, x_k] + \frac{(x_k - x_j)(f(x_l) - f(x_j))}{(x_k - x_j)(x_l - x_j)} - \frac{(x_l - x_j)(f(x_k) - f(x_j))}{(x_k - x_j)(x_l - x_j)} \\ = f[x_l, x_k] + \frac{f(x_l) - f(x_j)}{(x_l - x_j)} - \frac{f(x_k) - f(x_j)}{(x_k - x_j)} = f[x_l, x_k] + f[x_l, x_j] - f[x_k, x_j]. \end{aligned}$$

Llegando finalmente a lo que queríamos. Esta derivada nos dará dos opciones de aproximación. Luego tenemos una serie de posibilidades para la aproximación

- Aproximar $f[x_{i+1}, x^*]$ por la expresión $f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]$.
- Aproximar $f[x_{i+1}, x^*]$ por la expresión $f[x_{i+1}, x_i]$.
- Aproximar $f[x_{i+1}, x^*]$ por la expresión $f[x_{i+1}, x_{i-1}]$.

- Aproximar $f[x_{i-1}, x^*]$ por la expresión $f[x_{i+1}, x_{i-1}] + f[x_i, x_{i-1}] - f[x_{i+1}, x_{i-1}]$.
- Aproximar $f[x_{i-1}, x^*]$ por la expresión $f[x_{i+1}, x_{i-1}]$.
- Aproximar $f[x_{i-1}, x^*]$ por la expresión $f[x_i, x_{i-1}]$.

De esta manera en [5] definen 3 nuevos métodos

- Método 1: Aproxima $f[x_{i+1}, x^*]$ por $f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]$ y $f[x_{i-1}, x^*]$ por $f[x_{i+1}, x_{i-1}] + f[x_i, x_{i-1}] - f[x_{i+1}, x_{i-1}]$.

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_{i+1}, x_{i-1}] + f[x_i, x_{i-1}] - f[x_{i+1}, x_{i-1}]} \quad (3.11)$$

- Método 2: Aproxima $f[x_{i+1}, x^*]$ por $f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]$ y $f[x_{i-1}, x^*]$ por $f[x_{i+1}, x_{i-1}]$.

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_{i+1}, x_{i-1}]} \quad (3.12)$$

- Método 3: Aproxima $f[x_{i+1}, x^*]$ por $f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]$ y $f[x_{i-1}, x^*]$ por $f[x_i, x_{i-1}]$.

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_i, x_{i-1}]} \quad (3.13)$$

En [5] comentan como se plantearon otros dos nuevos métodos, uno que aproximaba $f[x_{i+1}, x^*]$ por $f[x_{i+1}, x_i]$ y $f[x_{i-1}, x^*]$ por $f[x_{i+1}, x_{i-1}] + f[x_i, x_{i-1}] - f[x_{i+1}, x_{i-1}]$ y el otro que aproximaba $f[x_{i+1}, x^*]$ por $f[x_{i+1}, x_i]$ y $f[x_{i-1}, x^*]$ por $f[x_{i+1}, x_{i-1}]$, aunque tras varios test numéricos se dieron cuenta de que no mejoraba nada, en muchos casos empeoraba de forma significativa, respecto al resto de métodos por lo tanto se descartaron y no se hizo un análisis de ellos, nosotros seguiremos la misma línea puesto que nuestro objetivo es presentar métodos cada vez mejores, los que empeoren nuestros resultados no nos interesan.

Para pasar ahora al análisis de la convergencia de estos métodos, en [5] consideran que x_{i+1} fue un paso sin modificar U y que el siguiente x_{i+2} será modificado. Introducen para esto varios conceptos que nos permitirán hacer el estudio de los tres métodos más sistemático. Si consideramos siempre que el paso anterior x_{i+1} fue sin modificar, tenemos por (2.8) y (3.2) que

$$\begin{aligned} \epsilon_{i+2} = x_{i+2} - x^* &\Rightarrow \epsilon_{i+2} = \frac{\gamma f_{i-1} x_{i+1} - f_{i+1} x_{i-1}}{\gamma f_{i-1} - f_{i+1}} - x^* = \frac{\gamma f_{i-1} \epsilon_{i+1} - f_{i+1} \epsilon_{i-1}}{\gamma f_{i-1} - f_{i+1}} \\ &= \left(\frac{\gamma f_{i-1} \epsilon_{i+1} - f_{i+1} \epsilon_{i-1}}{\gamma f_{i-1} - f_{i+1}} \right) \left(\frac{\epsilon_{i-1} \epsilon_{i+1}}{\epsilon_{i-1} \epsilon_{i+1}} \right) = \left(\frac{\epsilon_{i-1} \epsilon_{i+1}}{\gamma f_{i-1} - f_{i+1}} \right) \left(\frac{\gamma f_{i-1} \epsilon_{i+1} - f_{i+1} \epsilon_{i-1}}{\epsilon_{i-1} \epsilon_{i+1}} \right) \\ &= \left(\frac{\epsilon_{i-1} \epsilon_{i+1}}{\gamma f_{i-1} - f_{i+1}} \right) \left(\frac{\gamma f_{i-1}}{\epsilon_{i-1}} - \frac{f_{i+1}}{\epsilon_{i+1}} \right). \end{aligned}$$

Luego tenemos esta nueva forma de expresar el error del primer paso modificado

$$\epsilon_{i+2} = \frac{\epsilon_{i-1} \epsilon_{i+1}}{\gamma f_{i-1} - f_{i+1}} \left(\frac{\gamma f_{i-1}}{\epsilon_{i-1}} - \frac{f_{i+1}}{\epsilon_{i+1}} \right) \quad (3.14)$$

Notaremos,

$$\psi = \left(\frac{\gamma f_{i-1}}{\epsilon_{i-1}} - \frac{f_{i+1}}{\epsilon_{i+1}} \right). \quad (3.15)$$

Ya que realizábamos un primer paso sin modificar

$$\epsilon_{i+1} \sim \frac{c_2}{c_1} \epsilon_{i-1} \epsilon_i,$$

de (3.5) esto implica que, asintóticamente,

$$\text{signo}(\epsilon_{i+1}) = -\text{signo}\left(\frac{c_2}{c_1}\right), \quad (3.16)$$

solo hay que recordar (3.2), por lo tanto ϵ_i y ϵ_{i-1} tendrán signo contrario. De igual manera, y puesto que hemos dicho que el paso que produce ϵ_{i+2} era modificado, se evidencia que

$$\text{signo}(\epsilon_i) = -\text{signo}\left(\frac{c_2}{c_1}\right), \quad (3.17)$$

y

$$\text{signo}(\epsilon_{i-1}) = \text{signo}\left(\frac{c_2}{c_1}\right), \quad (3.18)$$

De otra manera la comprobación del tipo de paso $f_i f_{i+1}$ no nos daría mayor que cero, que es la condición para que ϵ_{i+2} provenga de un paso modificado.

Una vez visto todos estos conceptos vamos a aplicarlo a cada uno de los nuevos métodos.

Método 1

Recordando que γ está dada por (3.11), es decir,

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_{i+1}, x_{i-1}] + f[x_i, x_{i-1}] - f[x_{i+1}, x_{i-1}]}$$

En [5] muestra que la expresión (3.15) se comporta como

$$\psi \approx -c_2 \epsilon_{i-1},$$

y simplifican γ mediante la expresión

$$\gamma \approx \frac{c_1 + c_3 \epsilon_{i-1}^2 + 2c_2 \epsilon_{i+1}}{c_1 + 2c_2 \epsilon_i - 1 + c_3 \epsilon_{i-1} (2\epsilon_{i-1} + \epsilon_i)},$$

y sustituyendo esto en (3.14), asintóticamente

$$\epsilon_{i+2} \sim -\left(\frac{c_2}{c_1}\right) \epsilon_{i-1} \epsilon_{i+1}. \quad (3.19)$$

Vamos a ver que esto es así. Una vez hecha la sustitución para ψ realmente tenemos

$$\epsilon_{i+2} \sim \frac{\epsilon_{i-1} \epsilon_{i+1}}{\gamma f_{i-1} - f_{i+1}} (-c_2 \epsilon_{i-1}).$$

Aplicamos desarrollo de Taylor hasta la tercera derivada y sustituimos el valor de γ

$$\begin{aligned} \epsilon_{i+2} &\sim \frac{(\epsilon_{i-1}\epsilon_{i+1})(-c_2\epsilon_{i-1})}{\left(\frac{c_1+c_3\epsilon_{i-1}^2+2c_2\epsilon_{i+1}}{c_1+2c_2\epsilon_{i-1}+c_3\epsilon_{i-1}(2\epsilon_{i-1}+\epsilon_i)}\right)(c_1\epsilon_{i-1}+c_2\epsilon_{i-1}^2+c_3\epsilon_{i-1}^3)-(c_1\epsilon_{i+1}+c_2\epsilon_{i+1}^2+c_3\epsilon_{i+1}^3)} \\ &= \frac{(\epsilon_{i-1}\epsilon_{i+1})(-c_2\epsilon_{i-1})(c_1+2c_2\epsilon_{i-1}+c_3\epsilon_{i-1}(2\epsilon_{i-1}+\epsilon_i))}{(c_1+c_3\epsilon_{i-1}^2+2c_2\epsilon_{i+1})(c_1\epsilon_{i-1}+c_2\epsilon_{i-1}^2+c_3\epsilon_{i-1}^3)-(c_1\epsilon_{i+1}+c_2\epsilon_{i+1}^2+c_3\epsilon_{i+1}^3)(c_1+2c_2\epsilon_{i-1}+c_3\epsilon_{i-1}(2\epsilon_{i-1}+\epsilon_i))}. \end{aligned}$$

Como estamos estudiando su comportamiento asintótico suponemos que nos encontramos en un caso donde todo ϵ es menor que uno y nos quedamos únicamente con los factores mayores, es decir aquellos que contengan ϵ_{i-1} al menor exponente. Así reducimos la fracción

$$\epsilon_{i+2} \sim \frac{-c_2c_1\epsilon_{i-1}^2\epsilon_{i+1}}{c_1^2\epsilon_{i-1}} = -\left(\frac{c_2}{c_1}\right)\epsilon_{i-1}\epsilon_{i+1},$$

habiendo llegado a lo que queríamos probar.

Ahora, haciendo uso de (3.16) y (3.18), podemos deducir que en una vecindad de la raíz

$$\text{sign}(\epsilon_{i+2}) = -\text{sign}(\epsilon_{i+1}), \quad (3.20)$$

luego el paso siguiente lo obtendremos sin modificar. Trabajando de esta manera obtenemos un patrón de la forma AUU donde ahora A es el paso modificado. Veamos que esto es así, en este caso consideraremos $c_1 > 0$ pero $c_2 < 0$ de forma que estamos ante una función cóncava creciente por lo tanto al igual que antes si $x_i < x^*$ implica que $f_i < 0$ y $x_i > x^*$ implica que $f_i > 0 \forall i$. Luego por (3.17) y (3.18) tenemos $\epsilon_i > 0$ y $\epsilon_{i-1} < 0$ y así

$$\left. \begin{aligned} \epsilon_{i-1} < 0 &\Rightarrow x_{i-1} < x^* \Rightarrow f_{i-1} < 0 \\ \epsilon_i > 0 &\Rightarrow x_i > x^* \Rightarrow f_i > 0 \end{aligned} \right\}$$

luego $f_{i-1}f_i < 0$, tenemos un paso no modificado U.

$$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1}\right)\epsilon_i\epsilon_{i-1} \Rightarrow \epsilon_{i+1} > 0 \Rightarrow x_{i+1} > x^* \Rightarrow f_{i+1} > 0,$$

luego $f_i f_{i+1} > 0$, tenemos un paso modificado A.

$$\epsilon_{i+2} \sim -\left(\frac{c_2}{c_1}\right)\epsilon_{i-1}\epsilon_{i+1} \Rightarrow \epsilon_{i+2} < 0 \Rightarrow x_{i+1} < x^* \Rightarrow f_{i+1} < 0,$$

luego $f_{i+1}f_{i+2} < 0$, tenemos un paso no modificado U.

$$\epsilon_{i+3} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+1}\epsilon_{i+2} \Rightarrow \epsilon_{i+3} > 0 \Rightarrow x_{i+3} > x^* \Rightarrow f_{i+3} > 0,$$

luego $f_{i+2}f_{i+3} < 0$, tenemos un paso no modificado U.

▪

$$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+2} \epsilon_{i+3} \Rightarrow \epsilon_{i+4} > 0 \Rightarrow x_{i+4} > x^* \Rightarrow f_{i+4} > 0,$$

luego $f_{i+3}f_{i+4} > 0$, tenemos un paso modificado A .

▪

$$\epsilon_{i+5} \sim -\left(\frac{c_2}{c_1}\right) \epsilon_{i+2} \epsilon_{i+4} \Rightarrow \epsilon_{i+5} < 0 \Rightarrow x_{i+5} < x^* \Rightarrow f_{i+5} < 0,$$

luego $f_{i+4}f_{i+5} < 0$, tenemos un paso no modificado U .

▪

$$\epsilon_{i+6} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+4} \epsilon_{i+5} \Rightarrow \epsilon_{i+6} > 0 \Rightarrow x_{i+6} > x^* \Rightarrow f_{i+6} > 0,$$

luego $f_{i+5}f_{i+6} < 0$, tenemos un paso no modificado U .

▪

$$\epsilon_{i+7} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+5} \epsilon_{i+6} \Rightarrow \epsilon_{i+7} > 0 \Rightarrow x_{i+7} > x^* \Rightarrow f_{i+7} > 0,$$

luego $f_{i+6}f_{i+7} > 0$, tenemos un paso modificado A .

Y así podemos apreciar el patrón asintótico AUU del que hablábamos a partir de un primer paso no modificado U . De nuevo, la demostración es completamente análoga con cualquier combinación de signos para c_1 y c_2 respetando la norma que impusimos de que ϵ_{i+2} proviene de un modificado.

En [5] nos vuelven a mostrar su orden de eficiencia a partir de un patrón completo del método, obteniendo la relación

$$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right)^2 \epsilon_{i+1}^4 \epsilon_{i-2}^{-1}.$$

Así el orden que obtienen en [5] es $\rho \approx 3,73205$, y la eficiencia queda calculada al ser tres operaciones $\rho^{1/3} \approx 1,55113$.

Método 2

Ahora tenemos el γ de (3.12), es decir,

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_{i+1}, x_{i-1}]}.$$

Así, en [5] nos muestran que usando (3.15) llegamos a

$$\psi \approx -c_3 \epsilon_{i-1} \epsilon_i,$$

y simplifican γ mediante la expresión

$$\gamma \approx \frac{c_1 + 2c_2 \epsilon_{i+1} - c_3 \epsilon_{i-1} \epsilon_i}{c_1 + c_2(\epsilon_{i+1} + \epsilon_{i-1}) + c_3 \epsilon_{i-1}^2},$$

luego para un paso modificado usando (3.14) se llega

$$\epsilon_{i+2} \sim -\left(\frac{c_3}{c_1}\right)\epsilon_{i-1}\epsilon_i\epsilon_{i+1}. \quad (3.21)$$

Vamos a probarlo. De nuevo sustituyendo el valor de la ψ tenemos una expresión

$$\epsilon_{i+2} \sim \frac{\epsilon_{i-1}\epsilon_{i+1}}{\gamma f_{i-1} - f_{i+1}}(-c_3\epsilon_{i-1}\epsilon_i).$$

Aplicando desarrollo de Taylor hasta la tercera derivada y el valor de γ antes visto

$$\begin{aligned} \epsilon_{i+2} &\sim \frac{(\epsilon_{i-1}\epsilon_{i+1})(-c_3\epsilon_{i-1}\epsilon_i)}{\left(\frac{c_1+2c_2\epsilon_{i+1}-c_3\epsilon_{i-1}\epsilon_i}{c_1+c_2(\epsilon_{i+1}+\epsilon_{i-1})+c_3\epsilon_{i-1}^2}\right)(c_1\epsilon_{i-1}+c_2\epsilon_{i-1}^2+c_3\epsilon_{i-1}^3)-(c_1\epsilon_{i+1}+c_2\epsilon_{i+1}^2+c_3\epsilon_{i+1}^3)} \\ &= \frac{(\epsilon_{i-1}\epsilon_{i+1})(-c_3\epsilon_{i-1}\epsilon_i)(c_1+c_2(\epsilon_{i+1}+\epsilon_{i-1})+c_3\epsilon_{i-1}^2)}{(c_1+2c_2\epsilon_{i+1}-c_3\epsilon_{i-1}\epsilon_i)(c_1\epsilon_{i-1}+c_2\epsilon_{i-1}^2+c_3\epsilon_{i-1}^3)-(c_1\epsilon_{i+1}+c_2\epsilon_{i+1}^2+c_3\epsilon_{i+1}^3)(c_1+c_2(\epsilon_{i+1}+\epsilon_{i-1})+c_3\epsilon_{i-1}^2)}. \end{aligned}$$

Como en el anterior método, nos quedamos con los factores más grandes porque estamos ante un comportamiento asintótico

$$\epsilon_{i+2} \sim -\frac{c_3c_1\epsilon_{i-1}^2\epsilon_i\epsilon_{i+1}}{c_1^2\epsilon_{i-1}} = -\left(\frac{c_3}{c_1}\right)\epsilon_{i-1}\epsilon_i\epsilon_{i+1}.$$

Luego el estudio de este método se divide en dos partes, según como sea el signo de $\frac{c_3}{c_1}$.

Signo Negativo

Si tenemos que $\frac{c_3}{c_1} < 0$ las implicaciones (3.18),(3.17) y(3.16) nos llevan a que

$$\text{signo}(\epsilon_{i+2}) = -\text{signo}(\epsilon_{i+1}) = \text{signo}\left(\frac{c_2}{c_1}\right),$$

luego el siguiente paso será no modificado. Si en este caso denotamos un paso modificado por B , vamos a llegar asintóticamente a una secuencia de pasos de la forma EUU . Veamos nuevamente que se cumple, como en el anterior consideramos $c_1 > 0$ pero $c_2 < 0$ de forma que estamos ante una función cóncava creciente por lo tanto al igual que antes si $x_i < x^*$ implica que $f_i < 0$ y $x_i > x^*$ implica que $f_i > 0$, $\forall i$. Luego por (3.17) y (3.18) tenemos $\epsilon_i > 0$ y $\epsilon_{i-1} < 0$ y así

■

$$\left. \begin{aligned} \epsilon_{i-1} < 0 &\Rightarrow x_{i-1} < x^* \Rightarrow f_{i-1} < 0 \\ \epsilon_i > 0 &\Rightarrow x_i > x^* \Rightarrow f_i > 0 \end{aligned} \right\}$$

luego $f_{i-1}f_i < 0$, tenemos un paso no modificado U .

■

$$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1}\right)\epsilon_i\epsilon_{i-1} \Rightarrow \epsilon_{i+1} > 0 \Rightarrow x_{i+1} > x^* \Rightarrow f_{i+1} > 0,$$

luego $f_i f_{i+1} > 0$, tenemos un paso modificado B .

■

$$\epsilon_{i+2} \sim -\left(\frac{c_3}{c_1}\right)\epsilon_{i-1}\epsilon_i\epsilon_{i+1} \Rightarrow \epsilon_{i+2} < 0 \Rightarrow x_{i+1} < x^* \Rightarrow f_{i+1} < 0,$$

luego $f_{i+1}f_{i+2} < 0$, tenemos un paso no modificado U .

■

$$\epsilon_{i+3} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+1}\epsilon_{i+2} \Rightarrow \epsilon_{i+3} > 0 \Rightarrow x_{i+3} > x^* \Rightarrow f_{i+3} > 0,$$

luego $f_{i+2}f_{i+3} < 0$, tenemos un paso no modificado U .

■

$$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+2}\epsilon_{i+3} \Rightarrow \epsilon_{i+4} > 0 \Rightarrow x_{i+4} > x^* \Rightarrow f_{i+4} > 0,$$

luego $f_{i+3}f_{i+4} > 0$, tenemos un paso modificado B .

■

$$\epsilon_{i+5} \sim -\left(\frac{c_3}{c_1}\right)\epsilon_{i+2}\epsilon_{i+3}\epsilon_{i+4} \Rightarrow \epsilon_{i+5} < 0 \Rightarrow x_{i+5} < x^* \Rightarrow f_{i+5} < 0,$$

luego $f_{i+4}f_{i+5} < 0$, tenemos un paso no modificado U .

■

$$\epsilon_{i+6} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+4}\epsilon_{i+5} \Rightarrow \epsilon_{i+6} > 0 \Rightarrow x_{i+6} > x^* \Rightarrow f_{i+6} > 0,$$

luego $f_{i+5}f_{i+6} < 0$, tenemos un paso no modificado U .

■

$$\epsilon_{i+7} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+5}\epsilon_{i+6} \Rightarrow \epsilon_{i+7} > 0 \Rightarrow x_{i+7} > x^* \Rightarrow f_{i+7} > 0,$$

luego $f_{i+6}f_{i+7} > 0$, tenemos un paso modificado A .

Apreciando de forma inmediata el patrón BUU nuevamente tras un primer paso no modificado U . Análogo su desarrollo para otras variantes de signo de c_1 y c_2 mientras el resto de variables se ajusten para que el segundo paso sea modificado.

El índice de edificación computacional quedará de esta manera definido a partir de la secuencia que obtienen en [5]

$$\epsilon_{i+4} \sim \left(\frac{c_3}{c_1}\right)^2 \epsilon_{i+1}^5,$$

luego el orden es 5 a costa de 3 evaluaciones de la función, así el índice equivale a

$$5^{1/3} \approx 1,70998.$$

Signo Positivo

En este caso donde $\frac{c_3}{c_1} > 0$ las implicaciones (3.18),(3.17) y(3.16) nos llevan a que

$$\text{signo}(\epsilon_{i+2}) = \text{signo}(\epsilon_{i+1}) = -\text{signo}\left(\frac{c_2}{c_1}\right),$$

luego un nuevo paso modificado se generaría tras el primero. Para analizar este nuevo paso en [5] hacen referencia a una serie de definiciones originales del artículo [1], llegando finalmente a la expresión del error del siguiente paso modificado

$$\epsilon_{i+3} \approx \frac{c_2}{c_1} \epsilon_{i+1} \epsilon_{i+2}.$$

Así podemos llegar a una relación como la que encabezó esta sección

$$\text{signo}(\epsilon_{i+3}) = -\text{signo}(\epsilon_{i+2}) = \text{signo}\left(\frac{c_2}{c_1}\right).$$

Por lo que el paso siguiente será no modificado. Si aquí denotamos un paso modificado por C tenemos que asintóticamente el método seguirá una secuencia $CCUU$. Vamos a comprobarlo como en los casos pasados partiendo de las mismas condiciones, $c_1 > 0$, $c_2 < 0$, $\epsilon_i > 0$ y $\epsilon_{i-1} < 0$ teniendo

$$\left. \begin{array}{l} \epsilon_{i-1} < 0 \Rightarrow x_{i-1} < x^* \Rightarrow f_{i-1} < 0 \\ \epsilon_i > 0 \Rightarrow x_i > x^* \Rightarrow f_i > 0 \end{array} \right\}$$

luego $f_{i-1}f_i < 0$, tenemos un paso no modificado U .

$$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1}\right) \epsilon_i \epsilon_{i-1} \Rightarrow \epsilon_{i+1} > 0 \Rightarrow x_{i+1} > x^* \Rightarrow f_{i+1} > 0,$$

luego $f_i f_{i+1} > 0$, tenemos un primer paso modificado C .

$$\epsilon_{i+2} \sim -\left(\frac{c_3}{c_1}\right) \epsilon_{i-1} \epsilon_i \epsilon_{i+1} \Rightarrow \epsilon_{i+2} > 0 \Rightarrow x_{i+2} > x^* \Rightarrow f_{i+2} > 0,$$

luego $f_{i+1} f_{i+2} > 0$, tenemos un segundo paso modificado C .

$$\epsilon_{i+3} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+1} \epsilon_{i+2} \Rightarrow \epsilon_{i+3} < 0 \Rightarrow x_{i+3} < x^* \Rightarrow f_{i+3} < 0,$$

luego $f_{i+2} f_{i+3} < 0$, tenemos un paso no modificado U .

$$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+2} \epsilon_{i+3} \Rightarrow \epsilon_{i+4} > 0 \Rightarrow x_{i+4} > x^* \Rightarrow f_{i+4} > 0,$$

luego $f_{i+3} f_{i+4} < 0$, tenemos un paso no modificado U .

▪

$$\epsilon_{i+5} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+3} \epsilon_{i+4} \Rightarrow \epsilon_{i+5} > 0 \Rightarrow x_{i+5} > x^* \Rightarrow f_{i+5} > 0,$$

luego $f_{i+4}f_{i+5} > 0$, tenemos un primer paso modificado C.

▪

$$\epsilon_{i+6} \sim -\left(\frac{c_3}{c_1}\right) \epsilon_{i+3} \epsilon_{i+4} \epsilon_{i+5} \Rightarrow \epsilon_{i+6} > 0 \Rightarrow x_{i+6} > x^* \Rightarrow f_{i+6} > 0,$$

luego $f_{i+5}f_{i+6} > 0$, tenemos un segundo paso modificado C.

▪

$$\epsilon_{i+7} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+5} \epsilon_{i+6} \Rightarrow \epsilon_{i+7} < 0 \Rightarrow x_{i+7} < x^* \Rightarrow f_{i+7} < 0,$$

luego $f_{i+6}f_{i+7} < 0$, tenemos un paso no modificado U.

▪

$$\epsilon_{i+8} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+6} \epsilon_{i+7} \Rightarrow \epsilon_{i+8} > 0 \Rightarrow x_{i+8} > x^* \Rightarrow f_{i+8} > 0,$$

luego $f_{i+7}f_{i+8} < 0$, tenemos un paso no modificado U.

▪

$$\epsilon_{i+9} \sim \left(\frac{c_2}{c_1}\right) \epsilon_{i+7} \epsilon_{i+8} \Rightarrow \epsilon_{i+9} > 0 \Rightarrow x_{i+9} > x^* \Rightarrow f_{i+9} > 0,$$

luego $f_{i+8}f_{i+9} > 0$, tenemos un primer paso modificado C.

Nuestro patrón *CCUU* se cumple sin ningún problema, siendo otra vez análoga la demostración si variamos las condiciones que permitan que se cumpla que el segundo paso sea modificado.

En este caso la secuencia que nos aportan en [5] afirman que es de orden 8 mediante la expresión simplificada

$$\epsilon_{i+5} \sim -\left(\frac{c_2}{c_1}\right)\left(\frac{c_3}{c_1}\right)^3 \epsilon_{i+1}^8,$$

y podemos observar que son 4 evaluaciones de la función por lo tanto el índice de eficiencia computacional queda como

$$8^{1/4} \approx 1,68179.$$

Método 3

Para finalizar solo nos queda usar el γ proveniente de (3.13), es decir,

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_i, x_{i-1}]}.$$

Dándonos en [5] la última relación para (3.15)

$$\psi \approx -c_2 \epsilon_i,$$

a la vez que la última forma simplificada de γ

$$\gamma \approx \frac{c_1 + 2c_2\epsilon_{i+1}}{c_1 + c_2(\epsilon_{i-1} + \epsilon_i) + c_3(\epsilon_{i-1}^2 + \epsilon_{i-1}\epsilon_i + \epsilon_i^2)},$$

y, por (3.14), asintóticamente

$$\epsilon_{i+2} \sim -\left(\frac{c_2}{c_1}\right)\epsilon_i\epsilon_{i+1}. \quad (3.22)$$

Siguiendo en la misma línea de las otras pruebas, sustituimos el valor de ψ en la expresión general del error

$$\epsilon_{i+2} \sim \frac{\epsilon_{i-1}\epsilon_{i+1}}{\gamma f_{i-1} - f_{i+1}}(-c_2\epsilon_i).$$

Aplicando desarrollo de Taylor hasta la tercera derivada y el último valor de γ que nos han proporcionado

$$\begin{aligned} \epsilon_{i+2} &\sim \frac{(\epsilon_{i-1}\epsilon_{i+1})(-c_2\epsilon_i)}{\left(\frac{c_1 + 2c_2\epsilon_{i+1}}{c_1 + c_2(\epsilon_{i-1} + \epsilon_i) + c_3(\epsilon_{i-1}^2 + \epsilon_{i-1}\epsilon_i + \epsilon_i^2)}\right)(c_1\epsilon_{i-1} + c_2\epsilon_{i-1}^2 + c_3\epsilon_{i-1}^3) + (c_1\epsilon_{i+1} + c_2\epsilon_{i+1}^2 + c_3\epsilon_{i+1}^3)} \\ &= \frac{(\epsilon_{i-1}\epsilon_{i+1})(-c_2\epsilon_i)(c_1 + c_2(\epsilon_{i-1} + \epsilon_i) + c_3(\epsilon_{i-1}^2 + \epsilon_{i-1}\epsilon_i + \epsilon_i^2))}{(c_1 + 2c_2\epsilon_{i+1})(c_1\epsilon_{i-1} + c_2\epsilon_{i-1}^2 + c_3\epsilon_{i-1}^3) + (c_1\epsilon_{i+1} + c_2\epsilon_{i+1}^2 + c_3\epsilon_{i+1}^3)(c_1 + c_2(\epsilon_{i-1} + \epsilon_i) + c_3(\epsilon_{i-1}^2 + \epsilon_{i-1}\epsilon_i + \epsilon_i^2))}. \end{aligned}$$

Concentrándonos en su comportamiento asintótico se obtiene

$$\epsilon_{i+2} \sim \frac{-c_2c_1\epsilon_{i-1}\epsilon_i\epsilon_{i+1}}{c_1^2\epsilon_{i-1}} = -\left(\frac{c_2}{c_1}\right)\epsilon_i\epsilon_{i+1}.$$

Pudiendo afirmar por (3.17), (3.18) y (3.16) que

$$\text{signo}(\epsilon_{i+2}) = \text{signo}(\epsilon_{i+1}) = -\text{signo}\left(\frac{c_2}{c_1}\right),$$

así que será necesario un segundo paso modificado según este análisis. Como ocurrió anteriormente para poder efectuar este estudio adicional en [5] hacen uso de definiciones adicionales de [1], llegando finalmente a la expresión

$$\epsilon_{i+3} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+1}\epsilon_{i+2}\left(1 + 2\frac{\epsilon_i}{\epsilon_{i-1}}\right).$$

Como estamos estudiando un comportamiento asintótico podemos considerar que tenemos $\epsilon_i/\epsilon_{i-1} \rightarrow 0$, deduciendo que el error final puede ser

$$\epsilon_{i+3} \sim \alpha\epsilon_{i+1}\epsilon_{i+2},$$

para una cierta constante α que tiene el mismo signo que $\frac{c_2}{c_1}$. Con el estudio de los signos de los errores en el comienzo de este subapartado deducimos que ϵ_{i+3} y ϵ_{i+2} tendrán signos opuestos por lo tanto generaremos un paso no modificado. Si en este apartado final denotamos por D a un paso modificado, hemos deducido que asintóticamente seguirá una secuencia $DDUU$. Vamos a comprobarlo como en todos los casos, partiendo de las condiciones que cumplen que el segundo paso sea modificado $c_1 > 0$, $c_2 < 0$, $\epsilon_i > 0$ y $\epsilon_{i-1} < 0$ teniendo

$$\left. \begin{array}{l} \epsilon_{i-1} < 0 \Rightarrow x_{i-1} < x^* \Rightarrow f_{i-1} < 0 \\ \epsilon_i > 0 \Rightarrow x_i > x^* \Rightarrow f_i > 0 \end{array} \right\}$$

luego $f_{i-1}f_i < 0$, tenemos un paso no modificado U .

$$\epsilon_{i+1} \sim \left(\frac{c_2}{c_1}\right)\epsilon_i\epsilon_{i-1} \Rightarrow \epsilon_{i+1} > 0 \Rightarrow x_{i+1} > x^* \Rightarrow f_{i+1} > 0,$$

luego $f_i f_{i+1} > 0$, tenemos un primer paso modificado D .

$$\epsilon_{i+2} \sim -\left(\frac{c_2}{c_1}\right)\epsilon_i\epsilon_{i+1} \Rightarrow \epsilon_{i+2} > 0 \Rightarrow x_{i+2} > x^* \Rightarrow f_{i+2} > 0,$$

luego $f_{i+1}f_{i+2} > 0$, tenemos un segundo paso modificado D .

$$\epsilon_{i+3} \sim \alpha\epsilon_{i+1}\epsilon_{i+2} \Rightarrow \epsilon_{i+3} < 0 \Rightarrow x_{i+3} < x^* \Rightarrow f_{i+3} < 0,$$

luego $f_{i+2}f_{i+3} < 0$, tenemos un paso no modificado U .

$$\epsilon_{i+4} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+2}\epsilon_{i+3} \Rightarrow \epsilon_{i+4} > 0 \Rightarrow x_{i+4} > x^* \Rightarrow f_{i+4} > 0,$$

luego $f_{i+3}f_{i+4} < 0$, tenemos un paso no modificado U .

$$\epsilon_{i+5} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+3}\epsilon_{i+4} \Rightarrow \epsilon_{i+5} > 0 \Rightarrow x_{i+5} > x^* \Rightarrow f_{i+5} > 0,$$

luego $f_{i+4}f_{i+5} > 0$, tenemos un primer paso modificado D .

$$\epsilon_{i+6} \sim -\left(\frac{c_2}{c_1}\right)\epsilon_{i+4}\epsilon_{i+5} \Rightarrow \epsilon_{i+6} > 0 \Rightarrow x_{i+6} > x^* \Rightarrow f_{i+6} > 0,$$

luego $f_{i+5}f_{i+6} > 0$, tenemos un segundo paso modificado D .

$$\epsilon_{i+7} \sim \alpha\epsilon_{i+5}\epsilon_{i+6} \Rightarrow \epsilon_{i+7} < 0 \Rightarrow x_{i+7} < x^* \Rightarrow f_{i+7} < 0,$$

luego $f_{i+6}f_{i+7} < 0$, tenemos un paso no modificado U .

$$\epsilon_{i+8} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+6}\epsilon_{i+7} \Rightarrow \epsilon_{i+8} > 0 \Rightarrow x_{i+8} > x^* \Rightarrow f_{i+8} > 0,$$

luego $f_{i+7}f_{i+8} < 0$, tenemos un paso no modificado U .

$$\epsilon_{i+9} \sim \left(\frac{c_2}{c_1}\right)\epsilon_{i+7}\epsilon_{i+8} \Rightarrow \epsilon_{i+9} > 0 \Rightarrow x_{i+9} > x^* \Rightarrow f_{i+9} > 0,$$

luego $f_{i+8}f_{i+9} > 0$, tenemos un primer paso modificado C .

Hemos comprobado el patrón $CCUU$ tras un primer paso sin modificar. De esta manera en [5] llegan a deducir el orden de la expresión para una secuencia de este tipo

$$\epsilon_{i+9} \sim -\alpha \left(\frac{c_2}{c_1} \right)^4 \epsilon_{i+5}^7 \epsilon_{i+1}^{-1}.$$

Con esta ecuación en [5] obtienen que el orden es $\tau^4 \approx 6,85410$ donde $\tau = \frac{1+\sqrt{5}}{2}$. El índice de eficiencia computacional nos queda entonces

$$(\tau^4)^{1/4} = \tau \approx 1,61803.$$

Implementación en MATLAB[®]

Para finalizar este trabajo implementaremos los métodos vistos en el software científico MATLAB[®]. Para tener un resumen de todos los métodos a mano vamos a generar una tabla donde recordemos el patrón que genera cada uno y el índice de eficiencia computacional.

Método	Patrón Asintótico	Índice de Eficiencia
Illinois	IUU	1.44225
Pegasus	PPUU	1.64232
Método Ford 1	AUU	1.55113
Método Ford 2 ($c_3/c_1 < 0$)	BUU	1.70998
Método Ford 2 ($c_3/c_1 > 0$)	CCUU	1.68179
Método Ford 3	DDUU	1.61803

Cuadro 4.1: Tabla comparativa de los métodos

4.1 Código MATLAB[®]

En esta sección presentamos los códigos de invención propia para los diferentes seis métodos que hemos ido desarrollando en este trabajo en orden de aparición, *Regula Falsi*, *Método de Illinois*, *Método de Pegasus*, *Método de Ford 1*, *Método de Ford 2* y *Método de Ford 3*.

Cada programa nos pide una función que analizar f , un intervalo $[a, b]$ para comenzar el proceso iterativo en el cual ya nos aseguramos que se cumpla la condición del Teorema de Bolzano y un valor tol que será la tolerancia que vamos a considerar en cada problema, como criterio de parada usaremos que la evaluación del nuevo punto obtenido sea menor que nuestra tolerancia, es el criterio *ii*) de los vistos en (2.4). A su vez cada programa nos devolverá un valor aproximado para la raíz x , un *contador* que nos indicará la cantidad de iteraciones que ha hecho el programa para llegar a la raíz y también un array *respuestas* que contiene todas las aproximaciones de la raíz que el método obtiene en cada iteración que realiza.

Regula Falsi

Empezamos con el método inicial que encabeza el trabajo. Para este método, al ser poco fiable, le hemos añadido una condición de parada adicional, si ha realizado 500 iteraciones se detendrá y devolverá la raíz a la que haya podido llegar hasta ese momento.

```
function [x,contador,respuestas]=falsi(f,a,b,tol)
format long
fa=f(a);
fb=f(b);
contador=0;
```

```

respuestas=[];
fx=1;      %Para obligar a realizar una iteración
while abs(fx)>tol
    x=b-fb*(b-a)/(fb-fa);
    fx=f(x);
    if isnan(fx)      %Por si el resultado es tan pequeño que devuelva NaN
        fx=fb;
        x=b;
        break
    end
    if isnan(x)
        fx=fb;
        x=b;
        break
    end
end
if sign(fx) =sign(fb)      %Intervalo [x, b]
    a=x;
    fa=fx;
else      %Intervalo [a, x]
    b=x;
    fb=fx;
end
auxiliar=contador;
contador=auxiliar+1;
respuestas(contador)=x;
if contador >499      %Límite del número de iteraciones
    break
end
end
end

```

Método de Illinois

Ahora vemos el código correspondiente al *Método de Illinois*. Aquí ya vamos a apreciar la diferencia respecto a un paso no modificado y a uno modificado. Volvemos a colocar el limitador de iteraciones pero para asegurarnos del buen funcionamiento, ya veremos en las pruebas numéricas que no existe ningún problema con el número de iteraciones fuera de la *Regula Falsi*.

```

function [x,contador,respuestas]=illinois(f,a,b,tol)
fa=f(a);
fb=f(b);
contador=0;
respuestas=[];
fx=1;      %Para obligar a realizar una iteración
while abs(fx)>tol
    x=b-fb*(b-a)/(fb-fa);
    fx=f(x);

```

```

if isnan(fx)      %Por si el resultado es tan pequeño que devuelva NaN
    fx=fb;
    x=b;
    break
end
if isnan(x)
    fx=fb;
    x=b;
    break
end
if sign(fx) =sign(fb)      %Paso no modificado
    a=b;
    fa=fb;
else                      %Paso modificado
    fa=fa/2;
end
auxiliar=contador;
contador=auxiliar+1;
b=x;
fb=fx;
respuestas(contador)=x;
if contador >500
    break
end
end
end

```

Método de Pegasus

A continuación el código del *Método de Pegasus*. Ninguna diferencia respecto al *Método de Illinois* salvo en el paso modificado que posee una formulación diferente como ya estudiamos.

```

function [x,contador,respuestas]=pegasus(f,a,b,tol)
fa=f(a);
fb=f(b);
contador=0;
respuestas=[];
fx=1;      %Para obligar a realizar una iteración
while abs(fx)>tol
    x=b-fb*(b-a)/(fb-fa);
    fx=f(x);
    if isnan(fx)      %Por si el resultado es tan pequeño que devuelva NaN
        fx=fb;
        x=b;
        break
    end
end

```

```

if isnan(x)
    fx=fb;
    x=b;
    break
end
if sign(fx) =sign(fb)      %Paso no modificado
    a=b;
    fa=fb;
else                      %Paso modificado
    fa=(fa.*fb)/(fb+fx);
end
auxiliar=contador;
contador=auxiliar+1;
b=x;
fb=fx;
respuestas(contador)=x;
if contador >500
    break
end
end
end

```

Métodos de Ford

Ahora vamos a ver la implementación de los tres métodos de Ford. Recordaremos para cada uno cuál era el valor de la γ que les corresponde.

i) *Método de Ford 1*

Recordemos que en este método usamos

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_{i+1}, x_{i-1}] + f[x_i, x_{i-1}] - f[x_{i+1}, x_i]}$$

```

function [x,contador,respuestas]=metodo1(f,a,b,tol)
fa=f(a);
fb=f(b);
contador=0;
respuestas=[];
fx=1;      %Para obligar a realizar una iteración
while abs(fx)>tol
    x=b-fb*(b-a)/(fb-fa);
    fx=f(x);
    if isnan(fx)      %Por si el resultado es tan pequeño que devuelva NaN
        fx=fb;
        x=b;
        break
    end
end

```

```

    if isnan(x)
        fx=fb;
        x=b;
        break
    end
    if sign(fx) == sign(fb)      %Paso no modificado
        a=b;
        fa=fb;
    else                        %Paso modificado
        numerador=(fx-fb)/(x-b)+(fx-fa)/(x-a)-(fb-fa)/(b-a);
        denominador=(fx-fa)/(x-a)+(fb-fa)/(b-a)-(fx-fb)/(x-b);
        fa=fa.*(numerador./denominador);
    end
    auxiliar=contador;
    contador=auxiliar+1;
    b=x;
    fb=fx;
    respuestas(contador)=x;
    if contador > 500
        break
    end
end
end

```

ii) *Método de Ford 2*

En este método usamos

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_{i+1}, x_{i-1}]}$$

Aunque el análisis lo hayamos dividido según el signo de $\frac{c_3}{c_1}$, eso únicamente se tiene en cuenta a la hora del estudio teórico de los errores, de forma práctica el método es indiferente al signo, se implementa igual.

```

function [x,contador,respuestas]=metodo2(f,a,b,tol)
fa=f(a);
fb=f(b);
contador=0;
respuestas=[];
fx=1;      %Para obligar a realizar una iteración
while abs(fx)>tol
    x=b-fb*(b-a)/(fb-fa);
    fx=f(x);
    if isnan(fx)      %Por si el resultado es tan pequeño que devuelva NaN
        fx=fb;
        x=b;
        break
    end
    if isnan(x)

```

```

        fx=fb;
        x=b;
        break
    end
    if sign(fx) == sign(fb)      %Paso no modificado
        a=b;
        fa=fb;
    else                        %Paso modificado
        numerador=(fx-fb)/(x-b)+(fx-fa)/(x-a)-(fb-fa)/(b-a);
        denominador=(fx-fa)/(x-a);
        fa=fa*(numerador/denominador);
    end
    auxiliar=contador;
    contador=auxiliar+1;
    b=x;
    fb=fx;
    respuestas(contador)=x;
    if contador > 500
        break
    end
end
end

```

iii) *Método de Ford 3*

En este último método usamos

$$\gamma = \frac{f[x_{i+1}, x_i] + f[x_{i+1}, x_{i-1}] - f[x_i, x_{i-1}]}{f[x_i, x_{i-1}]}$$

```
function [x,contador,respuestas]=metodo3(f,a,b,tol)
```

```

fa=f(a);
fb=f(b);
contador=0;
respuestas=[];
fx=1;      %Para obligar a realizar una iteración
while abs(fx)>tol
    x=b-fb*(b-a)/(fb-fa);
    fx=f(x);
    if isnan(fx)      %Por si el resultado es tan pequeño que devuelva NaN
        fx=fb;
        x=b;
        break
    end
    if isnan(x)
        fx=fb;
        x=b;
        break
    end
end

```

```

if sign(fx) =sign(fb)      %Paso no modificado
    a=b;
    fa=fb;
else                      %Paso modificado
    numerador=(fx-fb)/(x-b)+(fx-fa)/(x-a)-(fb-fa)/(b-a);
    denominador=(fb-fa)/(b-a);
    fa=fa.*(numerador./denominador);
end
auxiliar=contador;
contador=auxiliar+1;
b=x;
fb=fx;
respuestas(contador)=x;
if contador >499
    break
end
end
end

```

4.2 Test Numéricos

Funciones del Test

Para las pruebas vamos a utilizar una serie de funciones tanto de elaboración propia como obtenidas de [5]. De [5] usaremos dos funciones:

Función de Ford 1.

$$\begin{aligned}
 f(x) &:= \frac{\exp(21000/x)}{(1,11 * 10^{11})x^2} - 1 \\
 x^* &= 551,77382493033
 \end{aligned}
 \tag{4.1}$$

Función de Ford 2.

$$\begin{aligned}
 f(x) &:= \frac{1}{x} + \ln(x) - 100 \\
 x^* &= 0,0095556044375379
 \end{aligned}
 \tag{4.2}$$

Ahora mostramos las 9 funciones de elaboración propia con las que trabajaremos. Las raíces a estas funciones han sido obtenidas mediante un calculador simbólico.

1.

$$f(x) := \frac{\sqrt{3x+10}+1}{2-\sqrt{x+3}} - 3$$
$$x^* = -2$$
(4.3)

2.

$$f(x) := \ln(x-5) + x^3 - \sqrt{77x^3+9} - 87$$
$$x^* = 6$$
(4.4)

3.

$$f(x) := \frac{\cos(x^3-37) + 3x^6 - 120x^3 + 337}{4x^4 + 7x^2 - 15x + 3}$$
$$x^* = 3,332221851645953$$
(4.5)

4.

$$f(x) := e^{(1/x^2+4x^3-2x+4)} - 1$$
$$x^* = -1,211003478827857$$
(4.6)

5.

$$f(x) := \frac{1513x^5 - 211883x^4 - 2375233x^3 + 17271395x^2 + 989100x - 12127500}{1523}$$
$$x^* = 150$$
(4.7)

6.

$$f(x) := \ln(x+3) - \frac{45}{(x-44)^3}$$
$$x^* = -2,000462195320353$$
(4.8)

7.

$$f(x) := e^{(3x-4)} - \log_{10}(144)x^2$$
$$x^* = 2,077097648088318$$
(4.9)

8.

$$f(x) := \cos\left(\frac{3}{x^2}\right) + \operatorname{sen}(\sqrt[5]{x}) - \Gamma(x)$$

$$x^* = 1,38726731574899$$
(4.10)

9.

$$f(x) := \ln\left(\tan\left(x - \frac{x^3}{2}\right)\right) + \operatorname{sen}(x)$$

$$x^* = 0,632074546692739$$
(4.11)

Resultados y Comparaciones

Una vez vistas las funciones que vamos a tratar y la implementación de los métodos en MATLAB[®], vamos a ver los resultados que obtenemos al usarlos. Daremos la raíz a la que llegan, el número de iteraciones que han necesitado y además la cadena de raíces que han obtenido antes de llegar a la deseada. Usaremos una tolerancia de 10^{-16} . Como la Regula Falsi ha tendido casi siempre a darnos una cantidad de iteraciones mayor que 500, marcaremos con un +500 cuando queramos decir que este ha sido el caso. Todos los cálculos numéricos se han efectuado en la versión MATLAB[®] R2018a y en un ordenador con Windows 10, un procesador Intel Core i5-7500T 2.70 GHz y 8 Gb de RAM.

En este ejemplo vamos a escribir todo el código usado para obtener los resultados de la primera función de elaboración propia. El cambio a cualquier otra función simplemente consiste en modificar la función que se define y el intervalo inicial de cada uno de los métodos.

```
clear all
format long
f=@(x) (sqrt(3.*x+10)+1)/(2-sqrt(x+3))-3;      %Definición de la Función
[x,y,z] = falsi(f,-3,-1,1/1000000);          %Método de la Regula Falsi
raiz(1,1)=x;
iteraciones(1,1)=y;
if y==500
    soluciones(1,1)=0;
else
for i=1:y
    soluciones(i,1)=z(i);
end
end
t=1:y;
plot(t,z(t),'*k-')
title('Regula Falsi')
xlabel('Número de iteraciones')
ylabel('Aproximación de la raíz')
figure
```

```
[x,y,z] = illinois(f,-3,-1,1/1000000);    %Método de Illinois
raiz(1,2)=x;
iteraciones(1,2)=y;
for i=1:y
    soluciones(i,2)=z(i);
end
t=1:y;
plot(t,z(t),'*r-')
title('Illinois')
xlabel('Número de iteraciones')
ylabel('Aproximación de la raíz')
figure
[x,y,z] = pegasus(f,-3,-1,1/1000000);    %Método de Pegasus
raiz(1,3)=x;
iteraciones(1,3)=y;
for i=1:y
    soluciones(i,3)=z(i);
end
t=1:y;
plot(t,z(t),'*c-')
title('Pegasus')
xlabel('Número de iteraciones')
ylabel('Aproximación de la raíz')
figure
[x,y,z] = metodo1(f,-3,-1,1/1000000);    %Método 1 de Ford
raiz(1,4)=x;
iteraciones(1,4)=y;
for i=1:y
    soluciones(i,4)=z(i);
end
t=1:y;
plot(t,z(t),'*b-')
title('Método 1')
xlabel('Número de iteraciones')
ylabel('Aproximación de la raíz')
figure
[x,y,z] = metodo2(f,-3,-1,1/1000000);    %Método 2 de Ford
raiz(1,5)=x;
iteraciones(1,5)=y;
for i=1:y
    soluciones(i,5)=z(i);
end
t=1:y;
plot(t,z(t),'*g-')
title('Método 2')
xlabel('Número de iteraciones')
ylabel('Aproximación de la raíz')
```

```

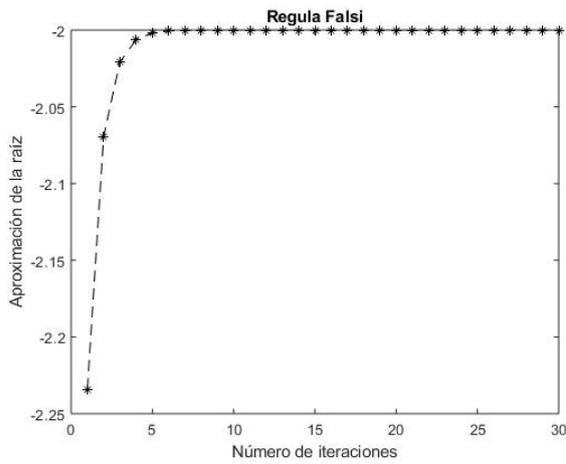
figure
[x,y,z] = metodo3(f,-3,-1,1/1000000);    %Método 3 de Ford
raiz(1,6)=x;
iteraciones(1,6)=y;
for i=1:y
    soluciones(i,6)=z(i);
end
t=1:y;
plot(t,z(t),'*m-')
title('Método 3')
xlabel('Número de iteraciones')
ylabel('Aproximación de la raíz')
disp('Raíz obtenida')
raiz
disp('Número iteraciones')
iteraciones
disp('Soluciones')
soluciones

```

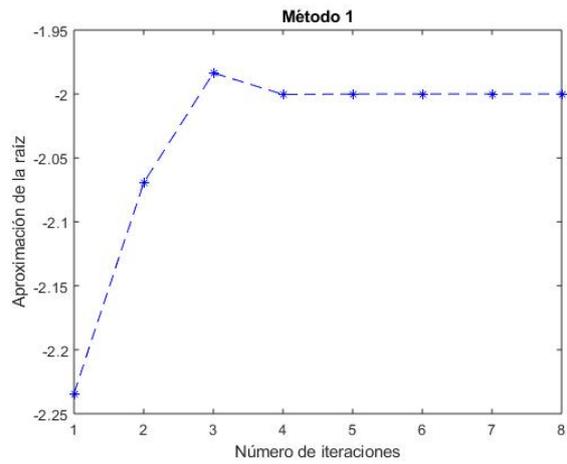
Ahora vamos a tabular las tres variables que nos das, siendo más destacada la cadena de raíces que efectúan los métodos antes de llegar a la que cumple la tolerancia.

Función 1	Regula Falsi	Illinois	Pegasus	Método 1	Método 2	Método 3
Intervalo: [-3,-1] $x^* = -2$	-2.234257304439173 -2.069419085259149 -2.020859163014927 -2.006290765648846 -2.001899207135381 -2.000573560500190 -2.000173231834045 -2.000052322527443 -2.000015803508440 -2.000004773307960 -2.000001441735998 -2.000000435463877 -2.000000131528104 -2.000000039726929 -2.000000011999176 -2.000000003624248 -2.000000001094673 -2.000000000330636 -2.000000000099866 -2.000000000030163 -2.000000000009110 -2.000000000002752 -2.000000000000831 -2.000000000000251 -2.000000000000076 -2.000000000000023 -2.000000000000007 -2.000000000000002 -2.000000000000001 -2.000000000000000	-2.234257304439173 -2.069419085259149 -1.976517685003626 -2.000438117205224 -2.000002796622208 -1.999997239740962 -2.00000000002091 -2.000000000000000	-2.234257304439173 -2.069419085259149 -2.006751518319917 -1.999987675220242 -2.00000022511650 -2.000000000000075 -2.000000000000000	-2.234257304439173 -2.069419085259149 -1.983556595647338 -2.000306426142903 -2.000001368183806 -1.99999993892423 -2.000000000000002 -2.000000000000000	-2.234257304439173 -2.069419085259149 -2.001860775311869 -1.999969108517589 -2.00000015563456 -2.000000000000131 -2.000000000000000	-2.234257304439173 -2.069419085259149 -2.004742920490924 -2.001899207135381 -2.00000053975623 -2.000000000000615 -2.000000000000000
Raíz obtenida	-2	-2	-2	-2	-2	-2
Número de Iteraciones	30	8	7	8	7	7

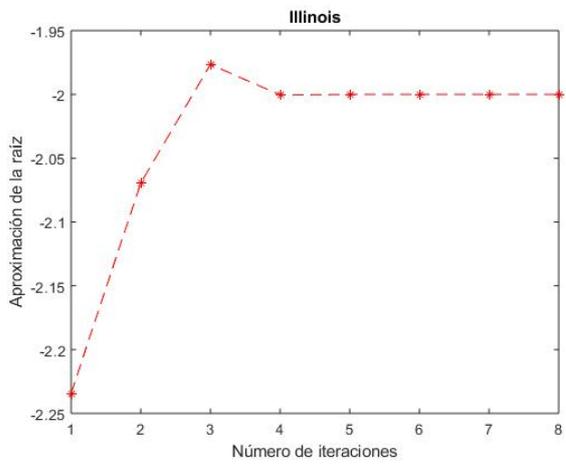
Para intentar ver más claramente como se aproximan los métodos a la raíz hagamos las gráficas de la iteración en la que estamos frente a lo que obtenemos en el método.



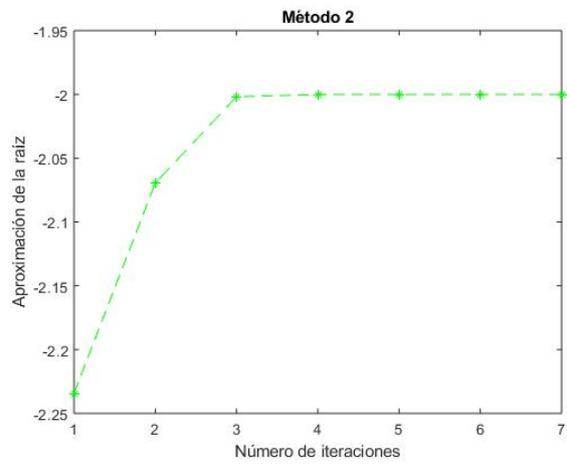
Gráfica de la Regula Falsi



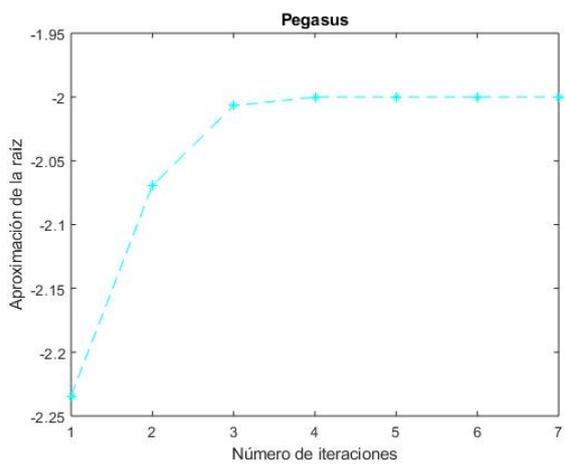
Gráfica del Método 1



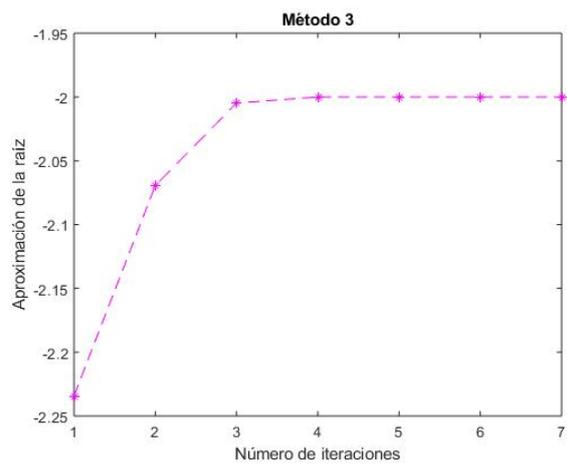
Gráfica de Illinois



Gráfica del Método 2



Gráfica de Pegasus



Gráfica del Método 3

El resto de tablas y gráficas correspondientes a las demás funciones se han añadido al apéndice de este trabajo para aclarar la lectura del mismo y su discusión la haremos en el siguiente capítulo.

Conclusiones

Una vez realizado todo el análisis de los diferentes métodos queda como último paso llegar a deducciones sobre los mismos. Lo primero que podemos ver de forma trivial es que la *Regula Falsi* es un método que ha quedado desfasado con respecto a sus modificaciones. Es evidente viendo la mayoría de los ejemplos en los cuales requiere de una cantidad enorme de iteraciones para llegar a una raíz que pueda tratarse de una buena aproximación pero que es peor que la del resto de métodos, que realizan una cantidad de iteraciones abrumadoramente menor. Aunque esto se suaviza si reducimos la tolerancia, pero es indiscutible el hecho de que la *Regula Falsi* obtiene peores resultados que el resto de métodos, además que no siempre nos podemos permitir una menor tolerancia para los objetivos que se plantean.

Si ahora nos centramos en los métodos que modifican la *Regula Falsi* podemos llegar a unas series de conclusiones. La primera es que el comportamiento de estos métodos se podría agrupar en dos grupos, el primero formado por *Illinois* y *Pegasus* y el segundo formado por los métodos de *Ford*.

El primer grupo posee un comportamiento muy similar, la mayor diferencia que se ha podido experimentar entre uno y otro es de dos iteraciones de diferencia en beneficio del *Método de Pegasus*, es decir, que o poseen la misma cantidad exacta de iteraciones o *Pegasus* llega a la raíz en dos iteraciones menos que el *Método de Illinois*, además en caso de no converger en menos de quinientas iteraciones lo harán los dos a la vez siempre pudiendo afirmar que de entre estos dos métodos nos quedaríamos con *Pegasus*, lo cual coincide con el hecho de que su eficiencia computacional es superior. Aquí también merece la pena apuntar el hecho de que si bien la *Regula Falsi* no da una aproximación excepcionalmente buena cuando se detiene por motivo de superar el límite de iteraciones, mientras que *Illinois* y *Pegasus* aunque se detengan por superar ese límite, la última aproximación es idéntica a la que llegan los *Métodos de Ford* cumpliendo el límite de iteraciones, por lo tanto podemos afirmar que aunque se detengan por motivos ajenos a cumplir con la tolerancia requerida obtenemos una aproximación buena. Se hace saber la posibilidad de que este problema no sea fruto del método sino del hecho que MATLAB[®] trabaja en doble precisión y hemos tomado una tolerancia muy pequeña (10^{-16}).

Hemos juntado todos los *Métodos de Ford* en un mismo grupo porque podemos apreciar por los ejemplos que tienen unos resultados similares. Además otro detalle a destacar son sus gráficas, no es apreciable en todas, pero si cogemos las correspondientes a las funciones 4, 5, 9 (véase el Apéndice que sigue a este capítulo) podemos ver claramente como este grupo de métodos tienen unas gráficas de aproximaciones de la raíz similares o al menos más parecidas entre sí que con los métodos de *Illinois* y *Pegasus*. Un detalle que tiene su origen en como surgieron los *Métodos de Ford* a partir de la misma idea de aproximar el γ idóneo (3.10) mientras que *Illinois* y *Pegasus* tienen un origen puramente experimental, tal es así que ni siquiera en el artículo de investigación del método de *Pegasus* explican de donde obtuvieron el valor de γ que presentan. Si ahora recogemos la información que tenemos sobre los *Métodos de Ford* podemos llegar a otra conclusión rápidamente, de entre los tres métodos, el que obtiene peores resultados es el *Método 1*, de hecho si volvemos a recurrir a las gráficas de los métodos, éste en particular obtiene unas que son las que difieren más respecto a la de los otros dos métodos. Este hecho concuerda de nuevo con la comparación de las eficiencias compu-

tacionales. En esta línea nos queda elegir entre el *Método 2* y el *Método 3* pero es una decisión difícil, no podemos encontrar diferencias en nuestras funciones salvo en la función 9 donde hay una iteración en beneficio del *Método 2*, pero esto no es significativo. Aunque un estudio más extenso ha dado como resultado que el *Método 2* muestra resultados ligeramente mejores, nuevamente del orden de una iteración o dos como máximo.

Como conclusión final solo quedaría comentar cual sería preferible a escoger, si *Pegasus* o el *Método 2*. Con la selección de funciones del Apéndice de este trabajo se puede ver rápidamente como la opción preferible final será el *Método 2*. No solo no cae en las +500 iteraciones donde *Pegasus* a veces incide si no que en los casos donde esto no le ocurre a *Pegasus* los resultados del *Método 2* son siempre mejores. Así podemos concluir que será preferible el *Método 2* frente a todos los demás.

Como todo método éste no carece de problemas, se hace una discusión de los mismos con dos ejemplos en la sección *Casos Patológicos* del Apéndice.

Bibliografía

- [1] N. Anderson, A. Björck. *A new high order method of Regula Falsi type for computing a root of an equation*, BIT, **13** (1973), 253-264.
- [2] U. M. Ascher, C. Greif. *A first course in numerical methods*, Society for Industrial and Applied mathematics, Vancouver, 2011.
- [3] M. Dowell, P. Jarratt. *A modified Regula Falsi method for computing the root of an equation*, BIT, **11** (1971), 168–174.
- [4] M. Dowell, P. Jarratt. *The “Pegasus” method for computing the root of an equation*, BIT, **12** (1972), 503–508.
- [5] J.A. Ford. *Improved Algorithms of Illinois-type for the numerical solution of nonlinear equations*, Technical Report, CSM-257. University of Essex, 1995.
- [6] A.S. Householder. *The numerical treatment of a single nonlinear equation*, McGraw-Hill, New York, 1970.
- [7] D. Kincaid. *Análisis numérico: las matemáticas del cálculo científico*, Addison-Wesley Iberoamericana, 1994.
- [8] A. Quarteroni, R. Sacco, F. Saleri. *Numerical mathematics*, Texts in applied mathematics 37. Springer, Milan, 2000.
- [9] J. F. Traub. *Iterative methods for the solution of equations*, Prentice-Hall, New York, 1964.

Apéndice

6.1 Tablas

Función 2

[5,5, 7]	Regula Falsi	Illinois	Pegasus	Método 1	Método 2	Método 3
Raíz obtenida	6	6	6	6	6	6
Número de Iteraciones	21	8	6	7	6	6

Función 3

[1, 100]	Regula Falsi	Illinois	Pegasus
Raíz obtenida	3.331927440990272	3.331927440990269	3.331927440990269
Número de Iteraciones	+500	11	11
	Método 1	Método 2	Método 3
Raíz obtenida	3.331927440990269	3.331927440990269	3.331927440990269
Número de Iteraciones	8	8	9

Función 4

[-2, 0,5]	Regula Falsi	Illinois	Pegasus
Raíz obtenida	-1.397000716257886	-1.211003478827858	-1.211003478827858
Número de Iteraciones	+500	+500	+500
	Método 1	Método 2	Método 3
Raíz obtenida	-1.211003478827858	-1.211003478827858	-1.211003478827858
Número de Iteraciones	21	17	17

Función 5

[100, 200]	Regula Falsi	Illinois	Pegasus	Método 1	Método 2	Método 3
Raíz obtenida	149.99999999999999	150	150	150	150	150
Número de Iteraciones	+500	13	11	15	13	14

Función 6

$[-2,9, -1]$	Regula Falsi	Illinois	Pegasus
Raíz obtenida	-2.000462195320353	-2.000462195320354	-2.000462195320354
Número de Iteraciones	+500	+500	+500
	Método 1	Método 2	Método 3
Raíz obtenida	-2.000462195320354	-2.000462195320354	-2.000462195320354
Número de Iteraciones	8	8	9

Función 7

$[0,5, 2]$	Regula Falsi	Illinois	Pegasus
Raíz obtenida	1.387267315748990	1.387267315748990	1.387267315748990
Número de Iteraciones	+500	+500	+500
	Método 1	Método 2	Método 3
Raíz obtenida	1.387267315748990	1.387267315748990	1.387267315748990
Número de Iteraciones	12	10	11

Función 8

$[1,5, 2,5]$	Regula Falsi	Illinois	Pegasus
Raíz obtenida	2.077097648088318	2.077097648088318	2.077097648088318
Número de Iteraciones	+500	11	10
	Método 1	Método 2	Método 3
Raíz obtenida	2.077097648088318	2.077097648088318	2.077097648088318
Número de Iteraciones	12	8	9

Función 9

$[0,1, 1]$	Regula Falsi	Illinois	Pegasus
Raíz obtenida	0.632074546692739	0.632074546692739	0.632074546692739
Número de Iteraciones	+500	10	8
	Método 1	Método 2	Método 3
Raíz obtenida	0.632074546692739	0.632074546692739	0.632074546692739
Número de Iteraciones	12	10	9

Función de Ford 1

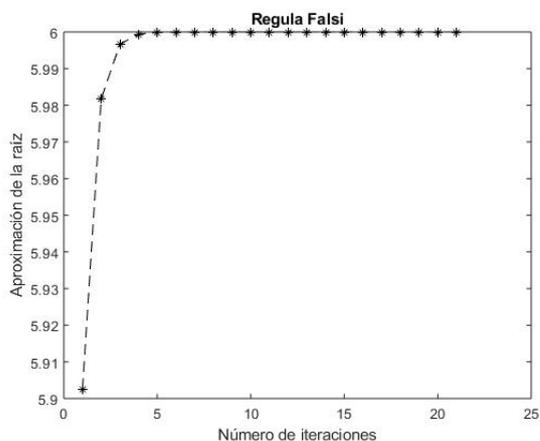
[400, 600]	Regula Falsi	Illinois	Pegasus
Raíz obtenida	599.9730008602588	551.7738249303267	551.7738249303266
Número de Iteraciones	+500	29	27
	Método 1	Método 2	Método 3
Raíz obtenida	551.7738249303267	551.7738249303266	551.7738249303267
Número de Iteraciones	15	12	12

Función de Ford 2

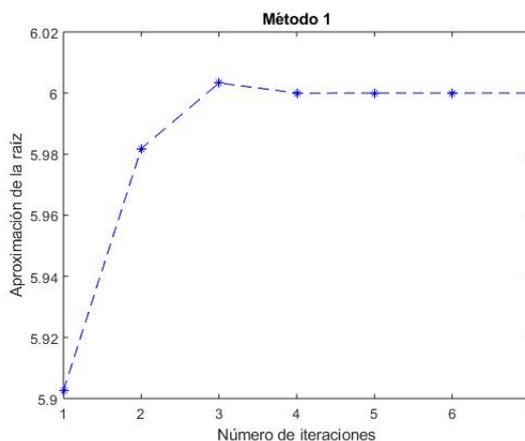
[0,0001, 0,1]	Regula Falsi	Illinois	Pegasus
Raíz obtenida	0.010060376564664	0.009555604437538	0.009555604437538
Número de Iteraciones	+500	17	16
	Método 1	Método 2	Método 3
Raíz obtenida	0.009555604437538	0.009555604437538	0.009555604437538
Número de Iteraciones	10	11	11

6.2 Gráficas

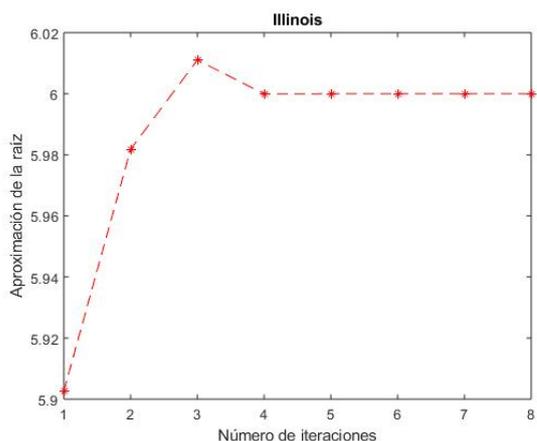
Función 2



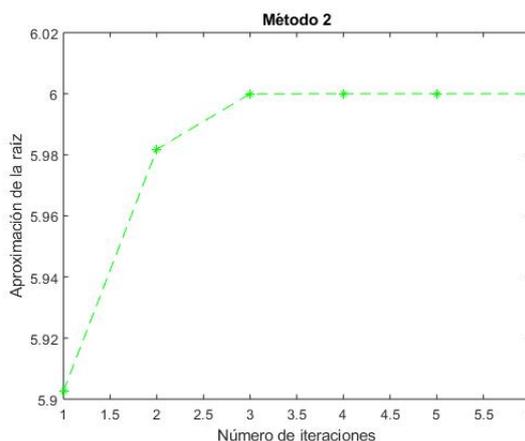
Gráfica de la Regula Falsi



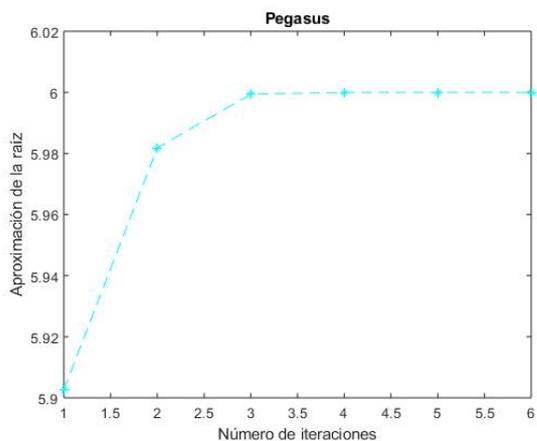
Gráfica del Método 1



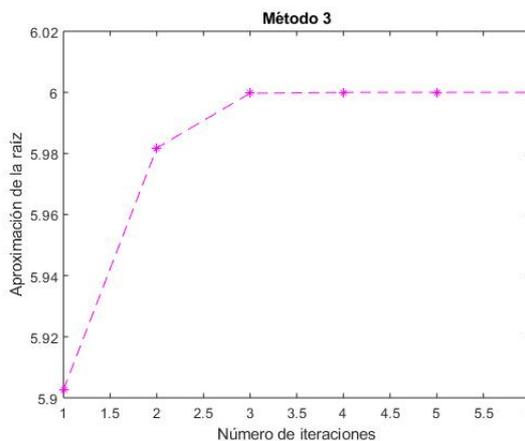
Gráfica de Illinois



Gráfica del Método 2

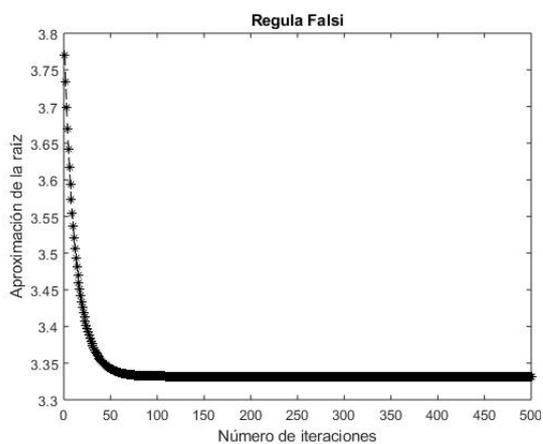


Gráfica de Pegasus

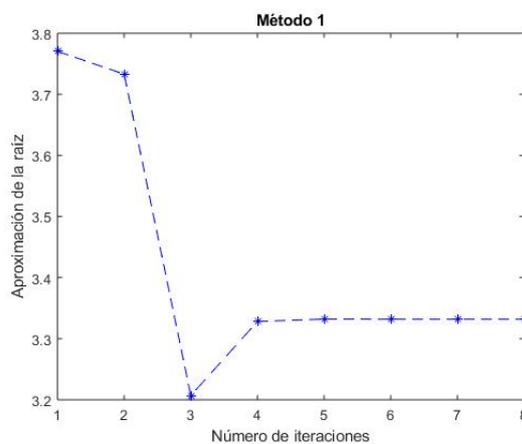


Gráfica del Método 3

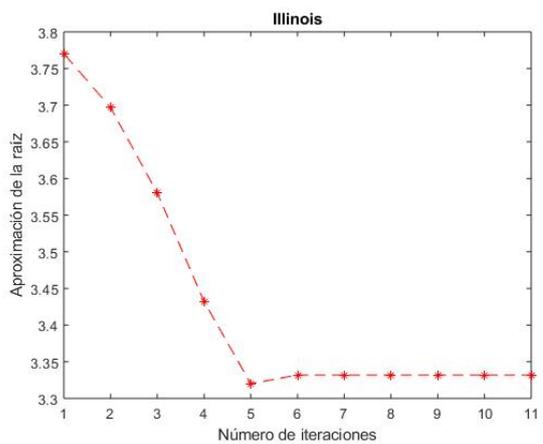
Función 3



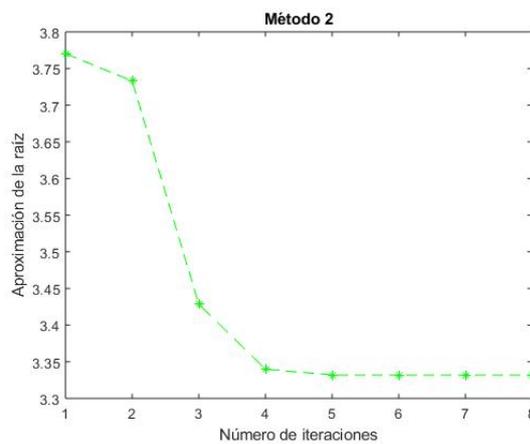
Gráfica de la Regula Falsi



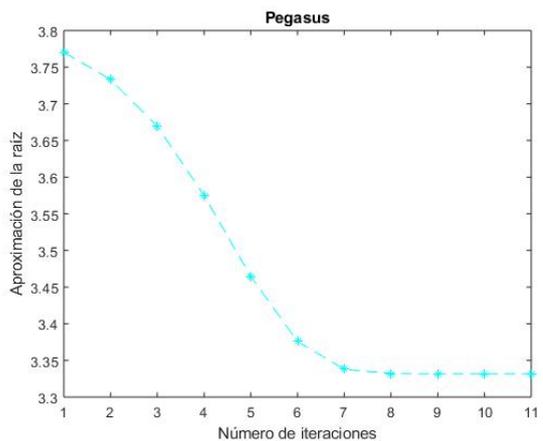
Gráfica del Método 1



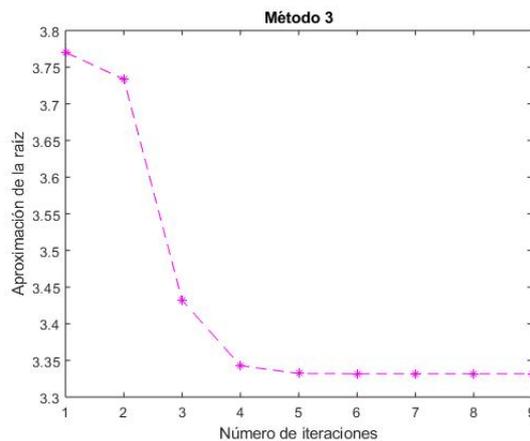
Gráfica de Illinois



Gráfica del Método 2

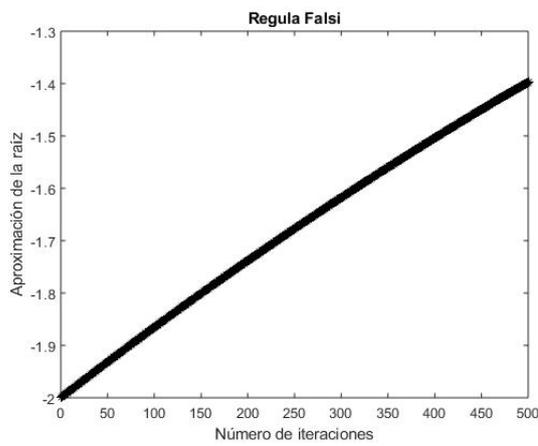


Gráfica de Pegasus

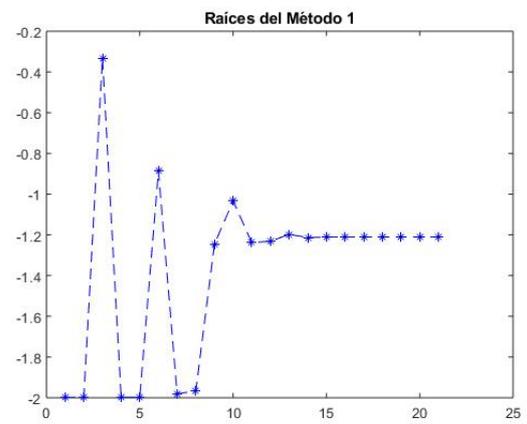


Gráfica del Método 3

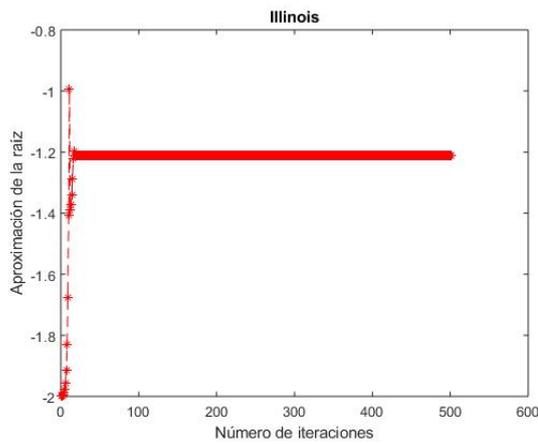
Función 4



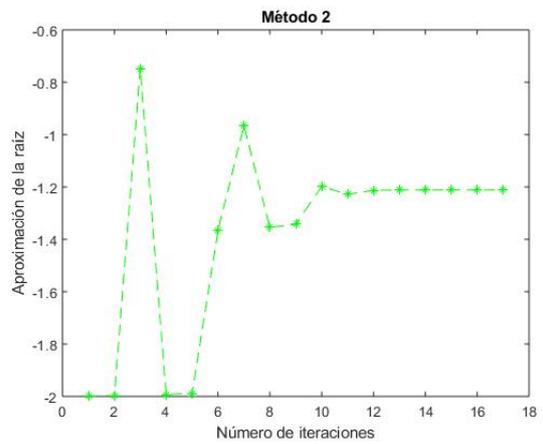
Gráfica de la Regula Falsi



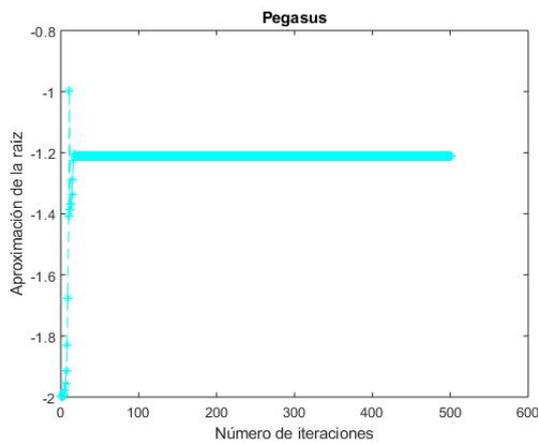
Gráfica del Método 1



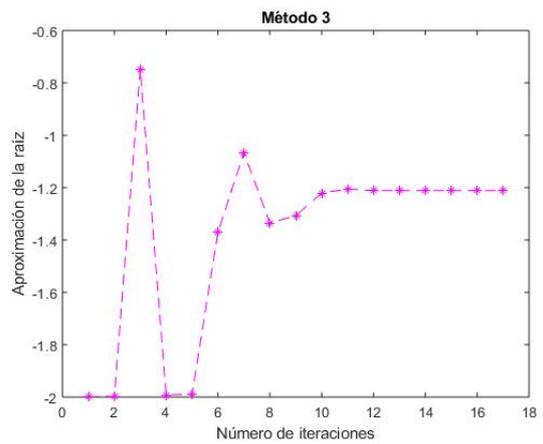
Gráfica de Illinois



Gráfica del Método 2

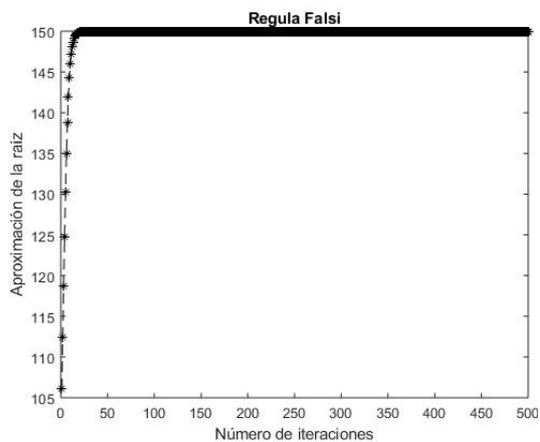


Gráfica de Pegasus

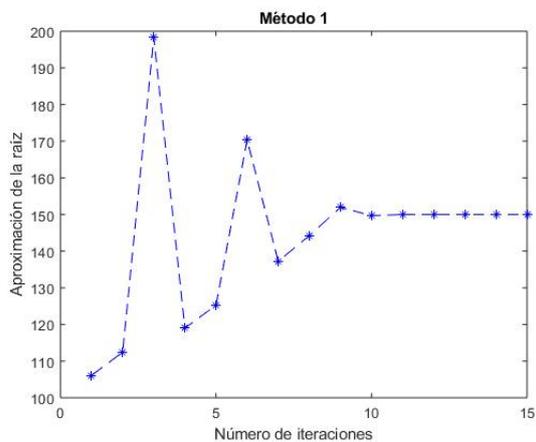


Gráfica del Método 3

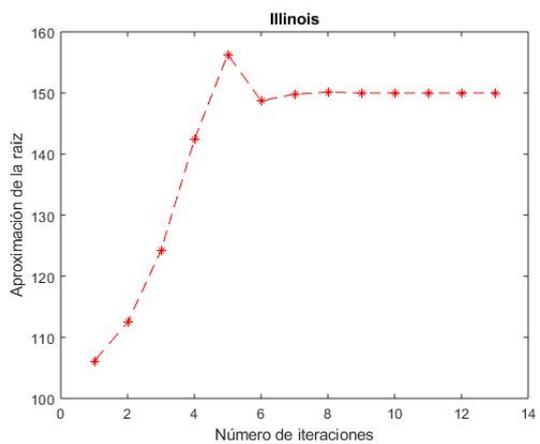
Función 5



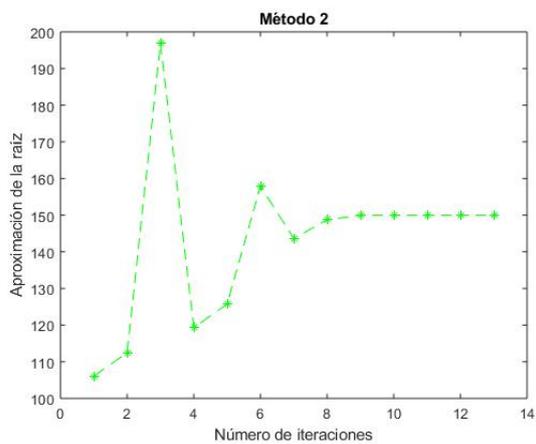
Gráfica de la Regula Falsi



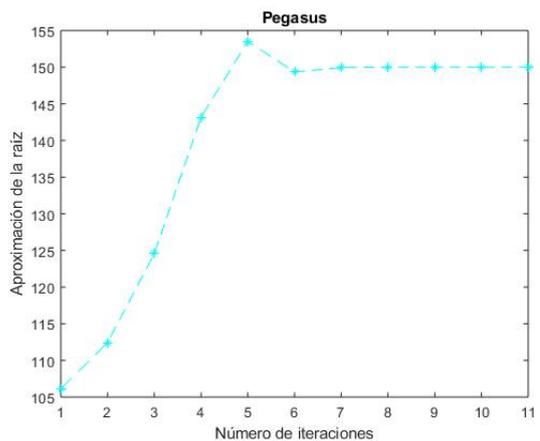
Gráfica del Método 1



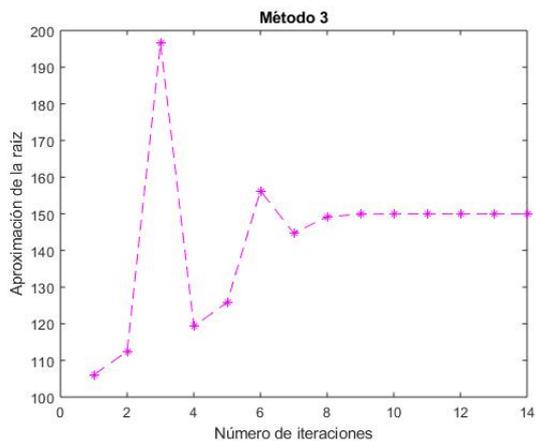
Gráfica de Illinois



Gráfica del Método 2

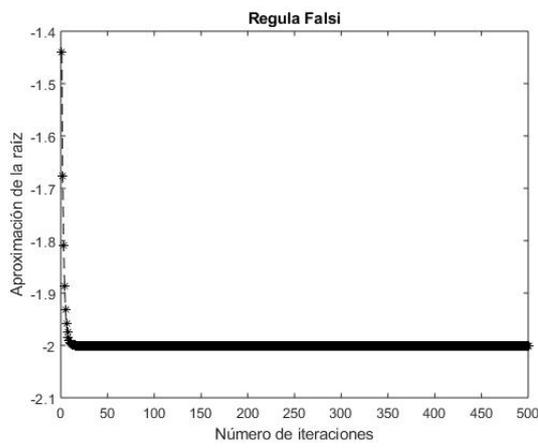


Gráfica de Pegasus

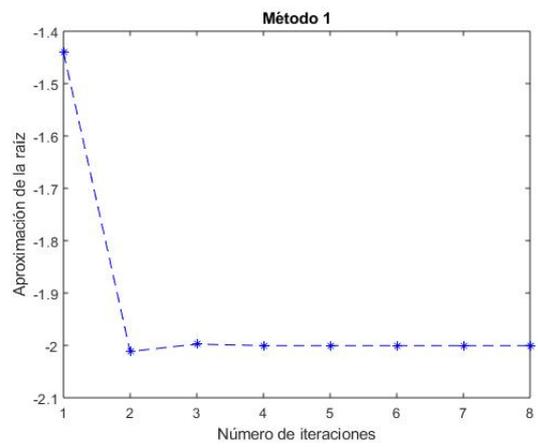


Gráfica del Método 3

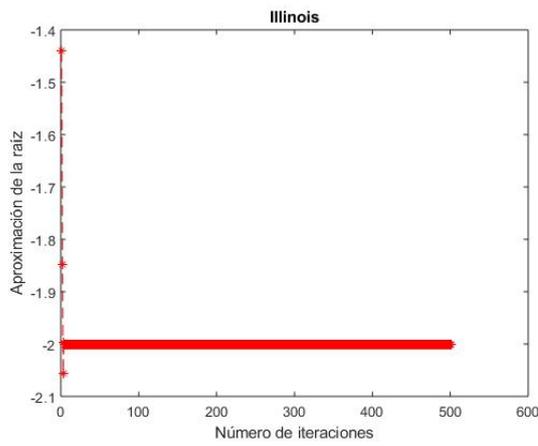
Función 6



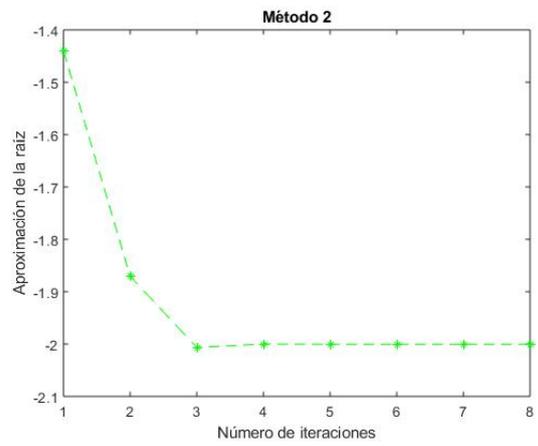
Gráfica de la Regula Falsi



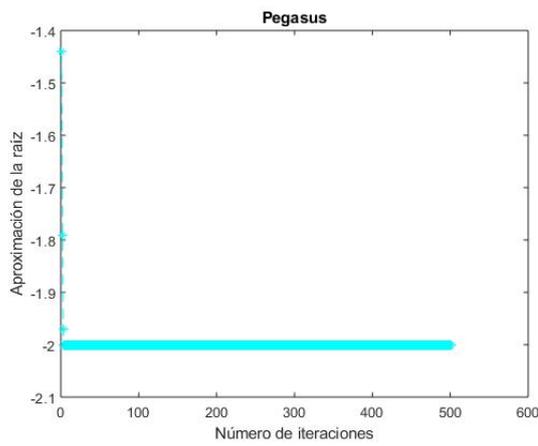
Gráfica del Método 1



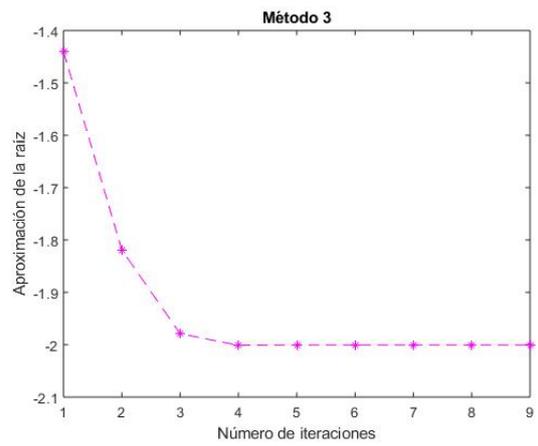
Gráfica de Illinois



Gráfica del Método 2

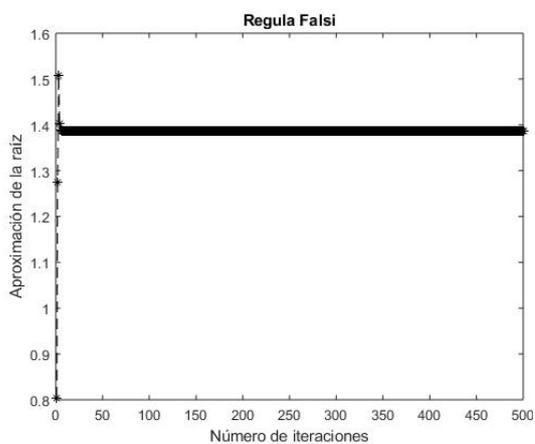


Gráfica de Pegasus

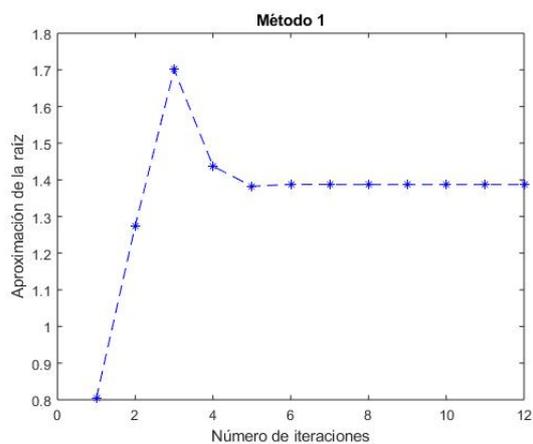


Gráfica del Método 3

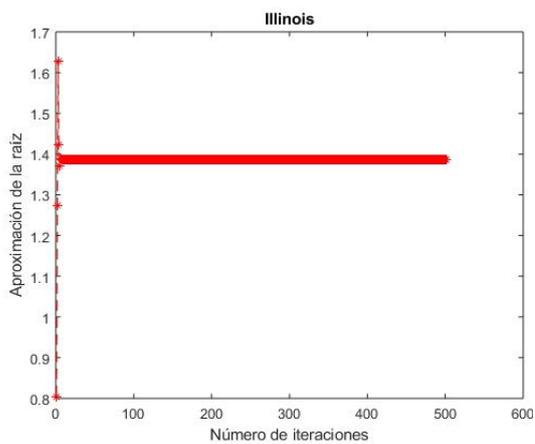
Función 7



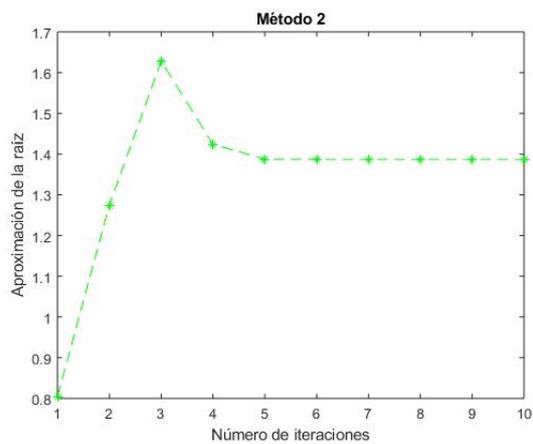
Gráfica de la Regula Falsi



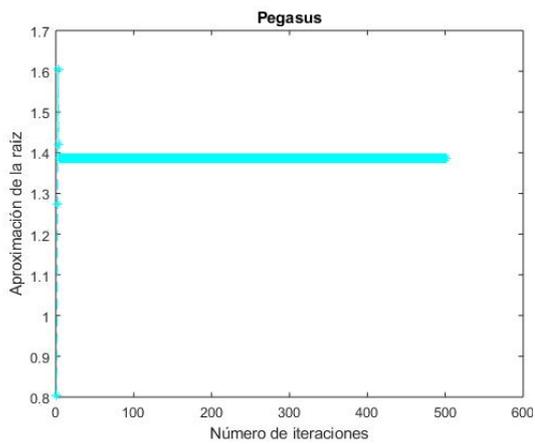
Gráfica del Método 1



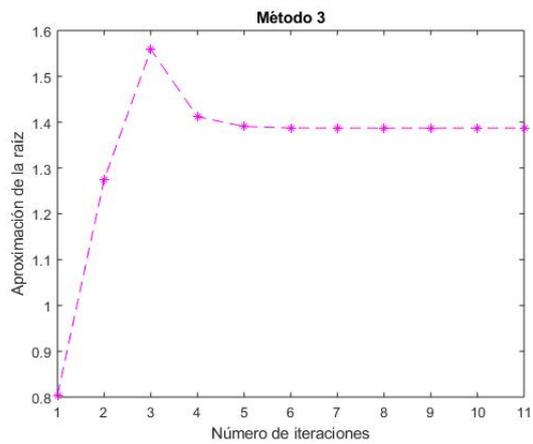
Gráfica de Illinois



Gráfica del Método 2

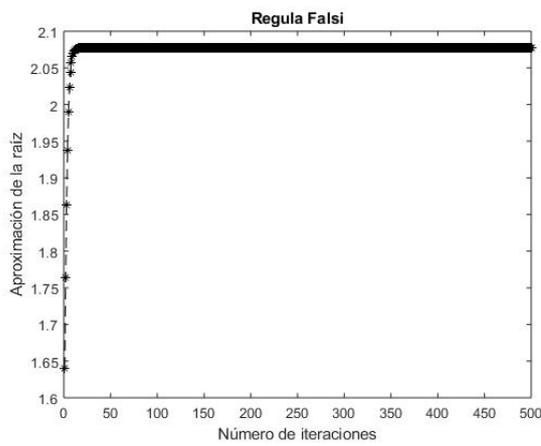


Gráfica de Pegasus

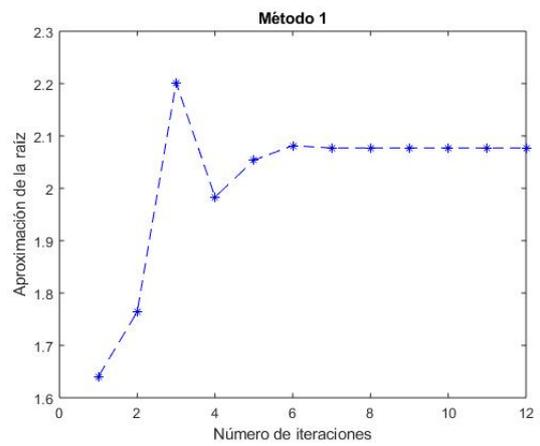


Gráfica del Método 3

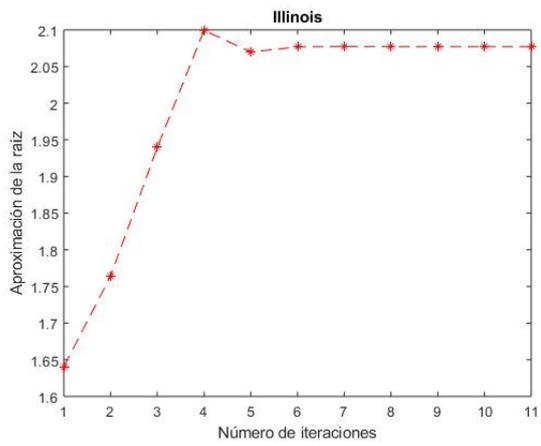
Función 8



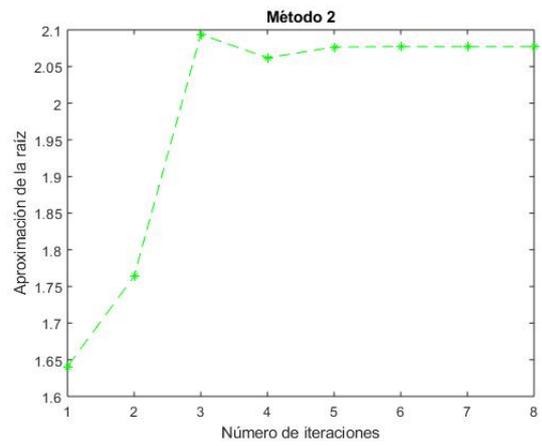
Gráfica de la Regula Falsi



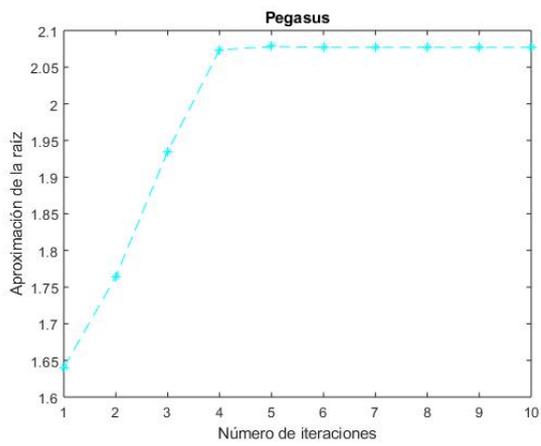
Gráfica del Método 1



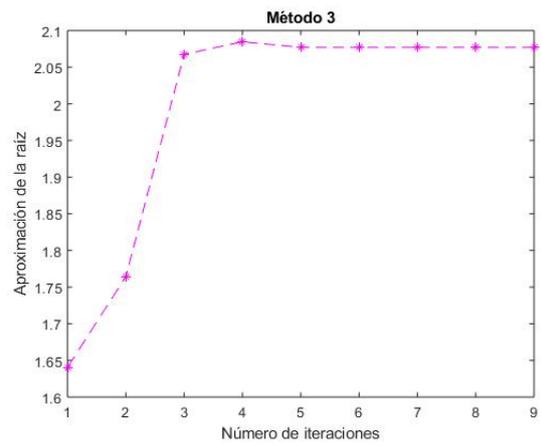
Gráfica de Illinois



Gráfica del Método 2

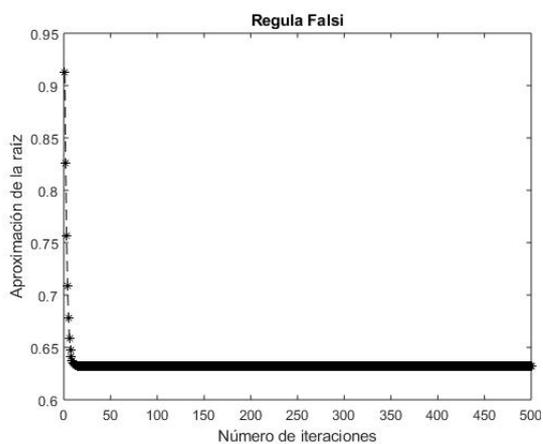


Gráfica de Pegasus

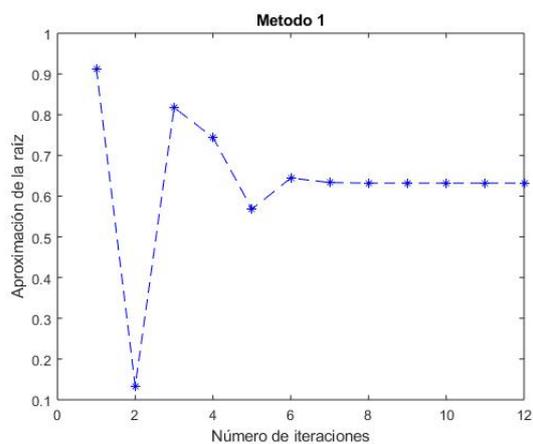


Gráfica del Método 3

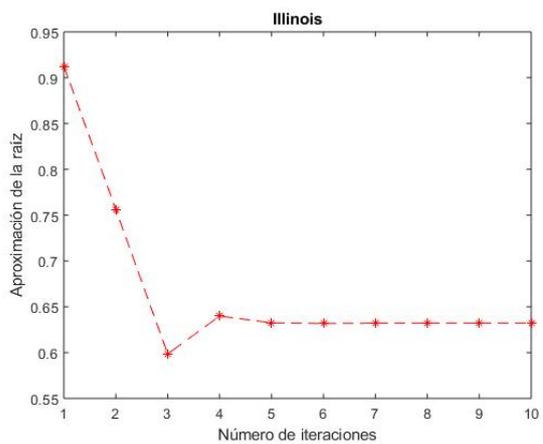
Función 9



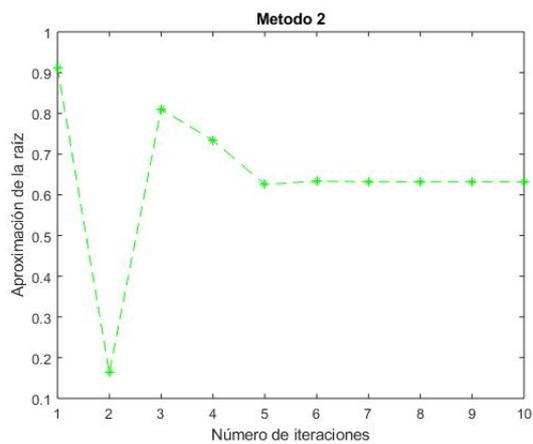
Gráfica de la Regula Falsi



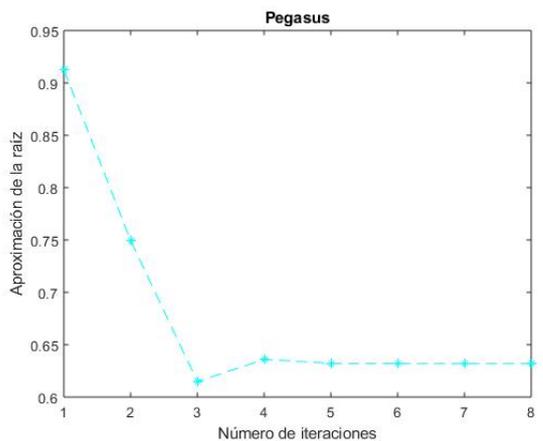
Gráfica del Método 1



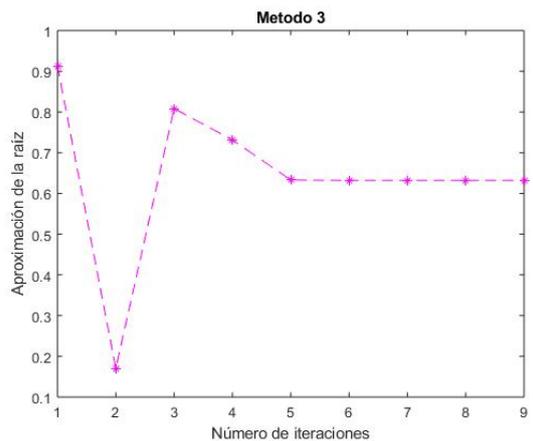
Gráfica de Illinois



Gráfica del Método 2

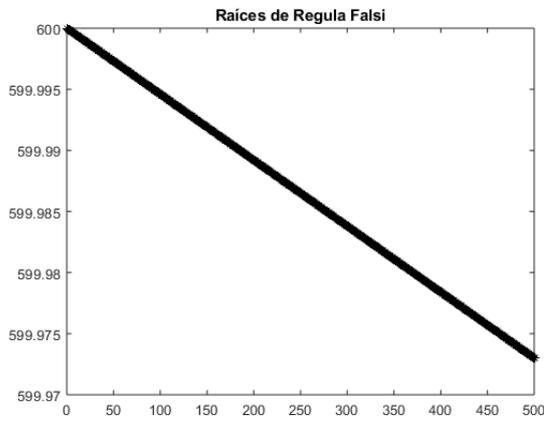


Gráfica de Pegasus

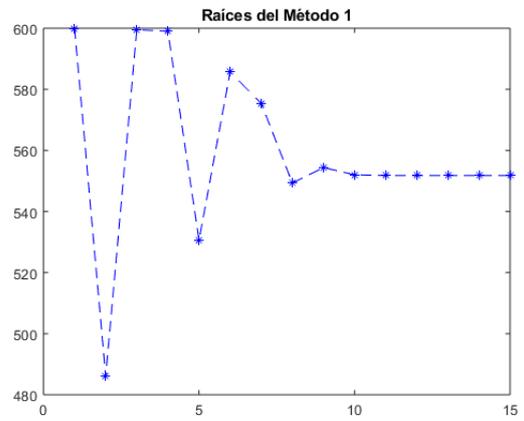


Gráfica del Método 3

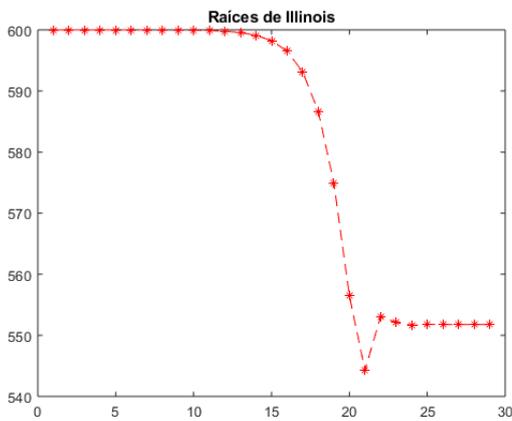
Función de Ford 1



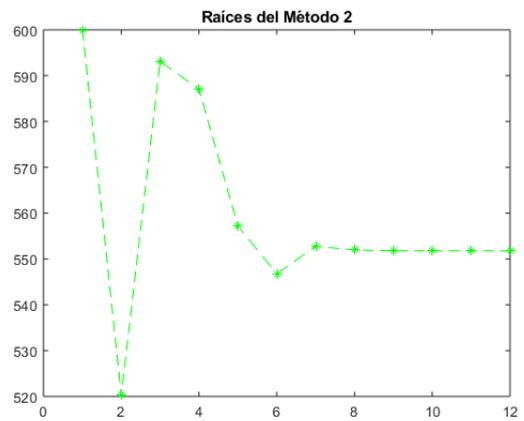
Gráfica de la Regula Falsi



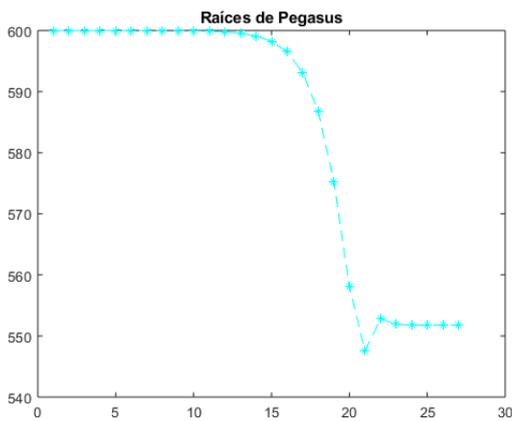
Gráfica del Método 1



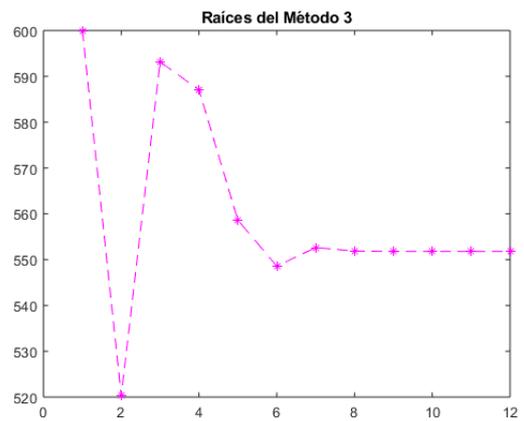
Gráfica de Illinois



Gráfica del Método 2

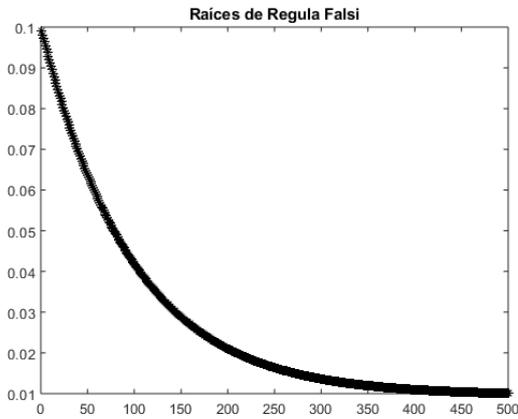


Gráfica de Pegasus

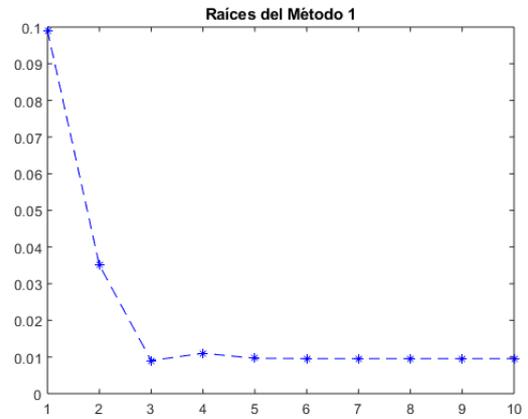


Gráfica del Método 3

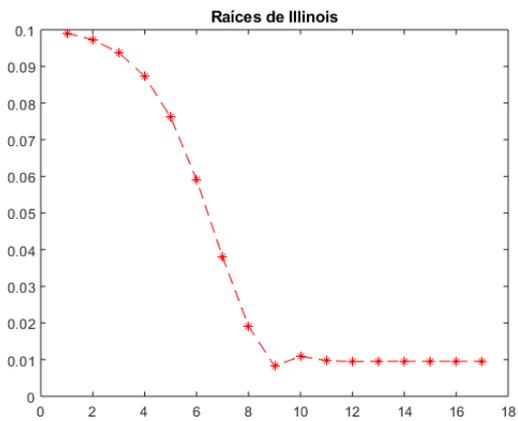
Función de Ford 2



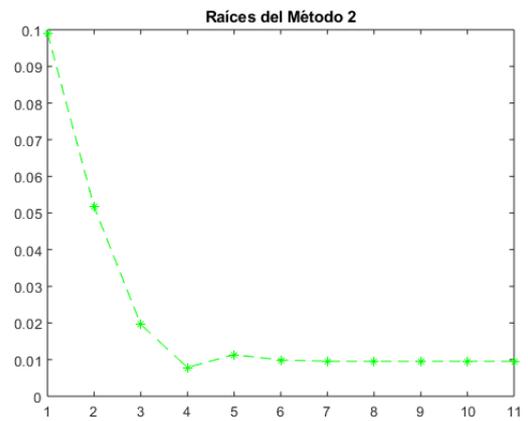
Gráfica de la Regula Falsi



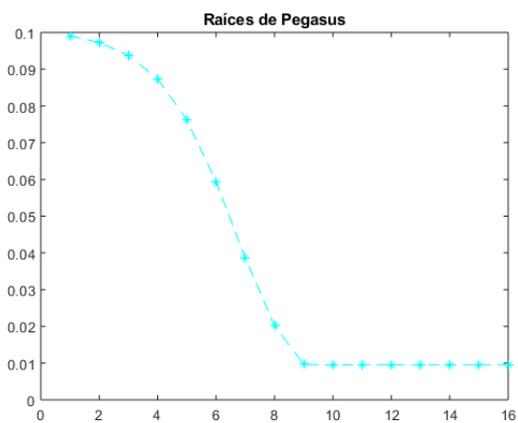
Gráfica del Método 1



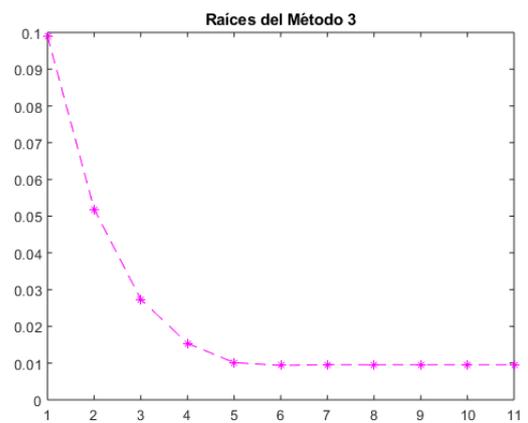
Gráfica de Illinois



Gráfica del Método 2



Gráfica de Pegasus



Gráfica del Método 3

6.3 Casos Patológicos

En esta sección presentamos dos problemas que hemos encontrado en los *Métodos de Ford* a raíz de las dos funciones sacadas de [5] que presentamos a continuación.

1.

$$f(x) := 4\cos(x) - \exp(x)$$

$$x^* = 0,90478821787302 \quad (6.1)$$

2.

$$f(x) := 10^{-8}(x-1) \prod_{i=1}^{10} (x^2 + x + i)$$

$$x^* = 1 \quad (6.2)$$

Como hemos hecho con el resto de funciones de esta sección, veamos sus tablas del número de iteraciones necesarias para cumplir la tolerancia establecida y la aproximación de la raíz a la que llegaron para luego observar las gráficas de las aproximaciones.

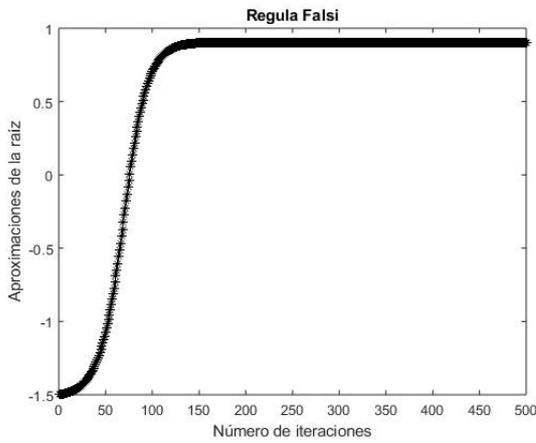
Caso 1

[0,1, 1]	Regula Falsi	Illinois	Pegasus
Raíz obtenida	0.904788217872992	0.904788217873019	0.904788217873019
Número de Iteraciones	+500	20	20
	Método 1	Método 2	Método 3
Raíz obtenida	-1.515864122805010	-1.515864122805010	-1.515864122805010
Número de Iteraciones	8	7	7

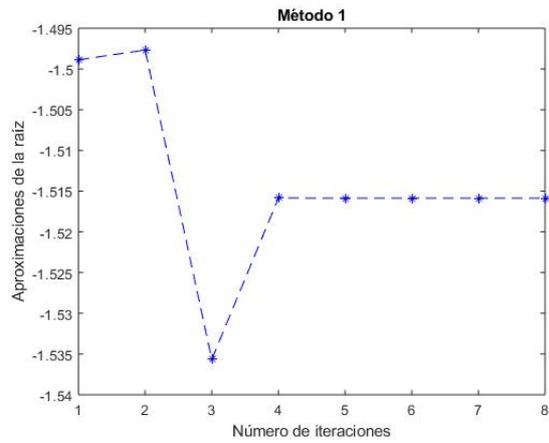
Caso 2

[0,5, 1,5]	Regula Falsi	Illinois	Pegasus	Método 1	Método 2	Método 3
Raíz obtenida	0.9999999999999999	1	1	1	1	1
Número de Iteraciones	+500	15	14	19	17	288

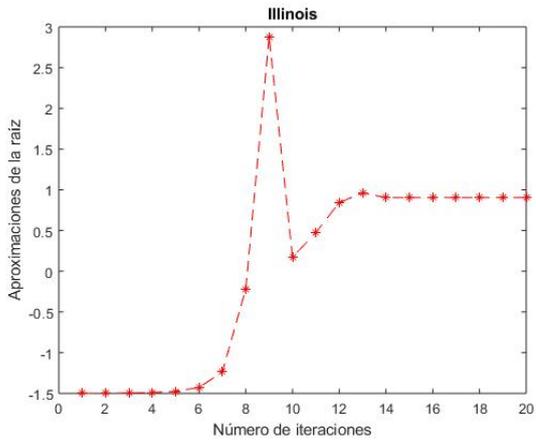
Caso 1



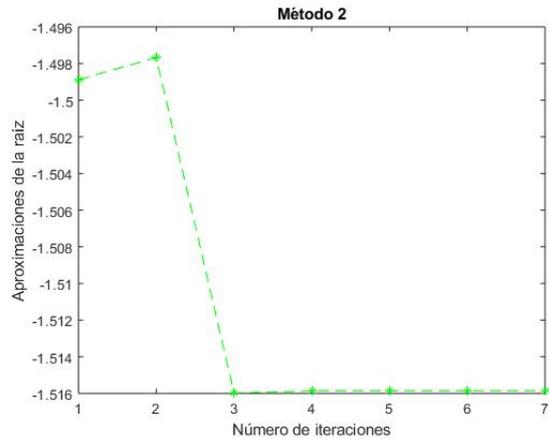
Gráfica de la Regula Falsi



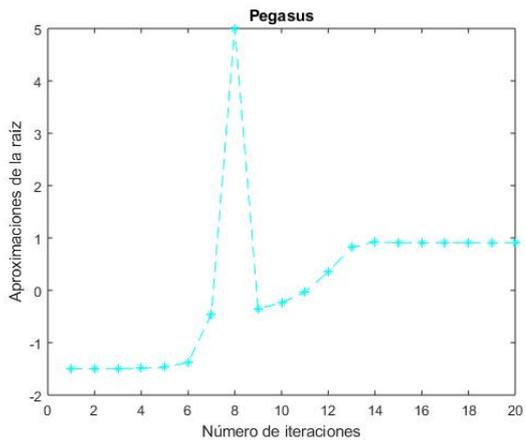
Gráfica del Método 1



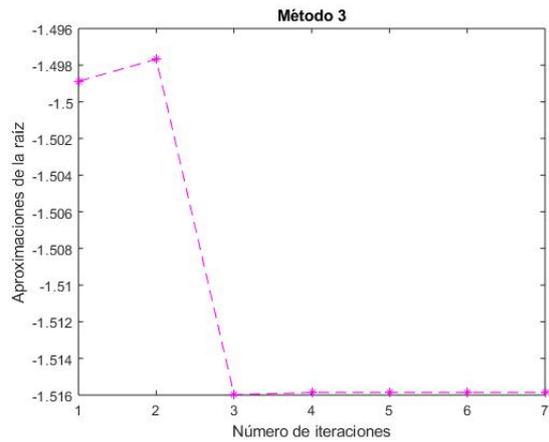
Gráfica de Illinois



Gráfica del Método 2

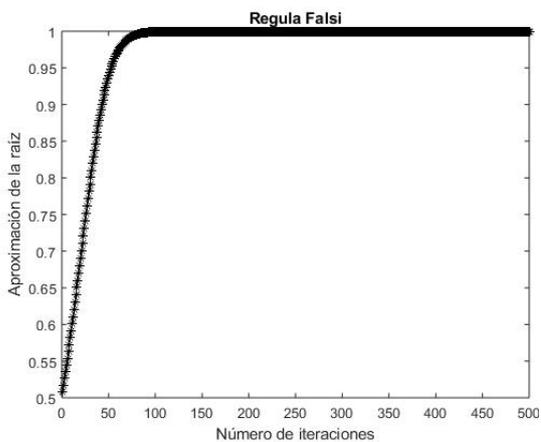


Gráfica de Pegasus

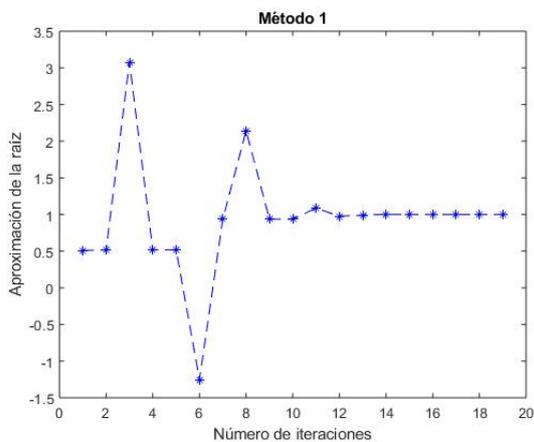


Gráfica del Método 3

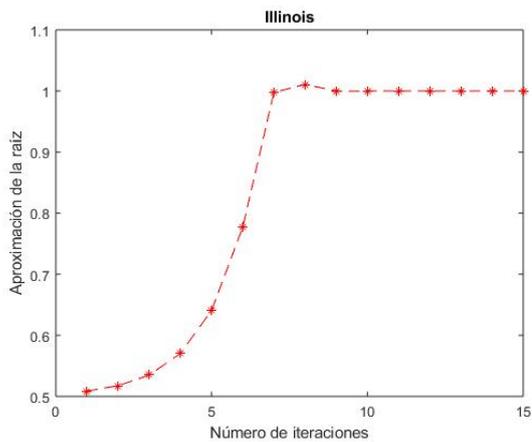
Caso 2



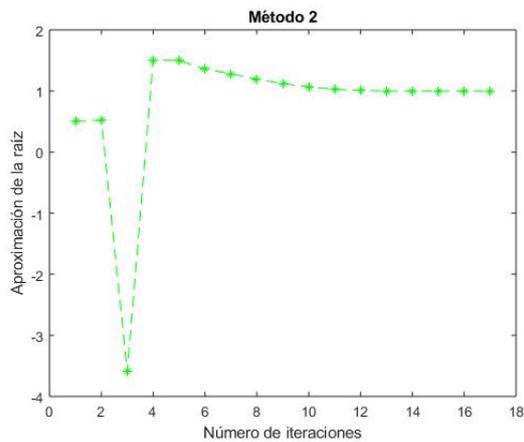
Gráfica de la Regula Falsi



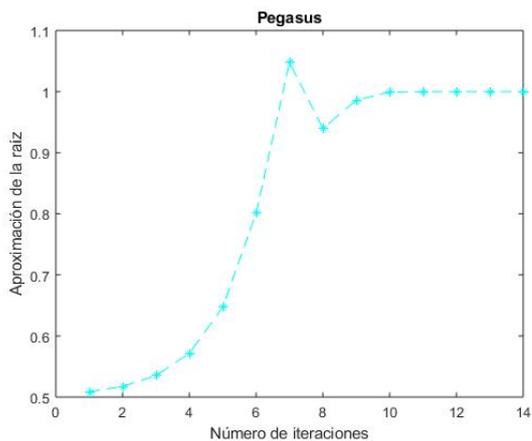
Gráfica del Método 1



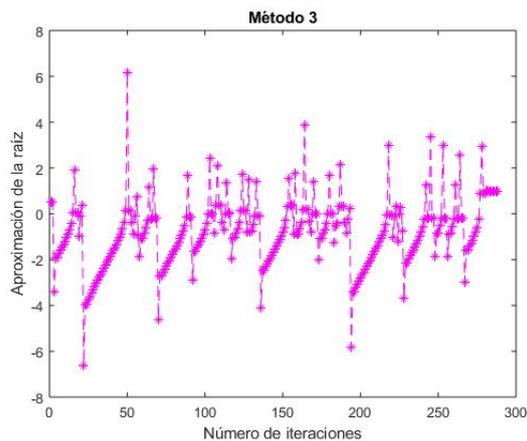
Gráfica de Illinois



Gráfica del Método 2



Gráfica de Pegasus



Gráfica del Método 3

Una vez visto esto podemos destacar los problemas que poseen estos métodos. El primero se puede apreciar en el *Caso 1*, nos encontramos con que los *Métodos de Ford* convergen a un número fuera del intervalo inicial pero que ese número también es una raíz, por lo tanto podemos afirmar que los métodos siguen funcionando aunque se llegue a una raíz no acotada inicialmente. Esto se puede deber a su forma de aproximar la raíz x^* pues al tomar como aproximación uno de los dos extremos, si hay una raíz fuera que se encuentra más cerca de uno de los extremos que la que se encuentra dentro del intervalo, la aproximación nos hará ir hacia ella. Si observamos las gráficas de las raíces podemos verlo, las de la *Regula Falsi*, *Illinois* y *Pegasus* en su proceso de llegar a la raíz siempre mantienen sus aproximaciones dentro del intervalo inicial mientras que los tres *Métodos de Ford* en el *Caso 1* y *Caso 2* podemos ver como toman aproximaciones de la raíz que se encuentran por debajo del extremo inferior del intervalo inicial, de hecho el *Método 1* también toma aproximaciones por encima del extremo superior del intervalo inicial en la *Caso 2*. El segundo problema que encontramos se encuentra únicamente en el *Caso 2* y más concretamente en el *Método 3* que podemos observar en su gráfica como oscila sin parar en busca de la aproximación de la raíz provocando que llegue a realizar 288 iteraciones cuando el resto de *Métodos de Ford* no llegan a 20, la explicación a este hecho se puede encontrar, como habíamos nombrado anteriormente, en la doble precisión que posee MATLAB[®] y podemos observar que cuando las aproximaciones se acercan mucho a cero las dos siguientes varían enormemente para volver a converger al cero hasta que finalmente avanza hacia el 1, pero aún con ello llega a la raíz correcta de forma exacta.

De esta forma podríamos volver a retomar la elección del mejor método sabiendo esto, nos encontramos con que del primer grupo es preferible *Pegasus* mientras que del segundo grupo es el *Método 2*. Atendiendo a estos casos que hemos denominado patológicos, podemos considerar al *Método de Pegasus* un método muy solvente.