



FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Resolución de ecuacións non lineares. Estratexias de globalización do método de Newton-Raphson

Lucía Revuelta Flores

2020/2021

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Resolución de ecuacións non lineares.
Estratexias de globalización do
método de Newton-Raphson

Lucía Revuelta Flores

Xullo, 2021

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Traballo proposto

Área de Coñecemento: Matemática Aplicada
Título: Resolución de ecuacións non lineares. Estratexias de globalización do método de Newton-Raphson.
Breve descrición do contido
Na materia de Cálculo Numérico nunha Variable estúdanse os métodos clásicos de dicotomía e o método de Newton-Raphson para aproximar as raíces dunha ecuación numérica non linear. Neste traballo trátase de estudar, ademais, os métodos da secante, regula falsi e o método de Müller. Na práctica, ningún destes métodos clásicos se usa de forma illada debido as súas debilidades, senón que se acompaña doutros métodos ou técnicas que lles aportan robustez e eficacia. En particular, estudaranse estratexias de globalización do método de Newton-Raphson.
Recomendacións
Ter cursado a materia de Cálculo Numérico nunha Variable e ter dominio na linguaxe de programación de Matlab.

Índice xeral

Resumo	VIII
Introdución	XI
1. Métodos clásicos	1
1.1. Método de dicotomía	1
1.1.1. Converxencia e estimación do erro	3
1.1.2. As vantaxes e inconvenientes	6
1.2. Método de regula falsi	7
1.2.1. As vantaxes e inconvenientes	8
1.3. Método de Newton-Raphson	10
1.3.1. Converxencia e erro	11
1.3.2. As vantaxes e inconvenientes	13
1.4. Método da secante	16
1.4.1. As vantaxes e inconvenientes	19
1.5. Método de Müller	21
1.5.1. As vantaxes e inconvenientes	24
2. Estratexias de globalización do método de Newton-Raphson	27
2.1. Método híbrido de Dicotomía e Newton-Raphson	27
2.1.1. Resultados	30
2.1.2. Conclusións	33
2.2. Método híbrido: dicotomía e regula falsi	33
2.2.1. Resultados	35
2.2.2. Conclusións	37
2.3. Método de Dekker-Brent	37
2.3.1. Interpolación cadrática inversa	37
2.3.2. Algoritmo de Dekker-Brent	39

2.3.3. Conclusións	44
Apéndice	47
A. Código Matlab para os métodos clásicos	47
A.1. Método de dicotomía	47
A.2. Método de regula falsi	49
A.3. Método de Newton-Raphson	50
A.4. Método da secante	51
A.5. Método de Müller	52
B. Código Matlab para os métodos híbridos	55
B.1. Método híbrido de dicotomía e Newton-Raphson	55
B.2. Método híbrido de dicotomía e regula falsi	57
B.3. Método de Dekker-Brent	59
Bibliografía	61

Resumo

No primeiro capítulo deste traballo introducimos os métodos clásicos de dicotomía, regula falsi, Newton-Raphson, secante e Müller para a resolución de ecuacións non lineares nunha variable. Na práctica, ningún destes métodos se presenta de forma illada debido as súas problemáticas, senón que se acompañan doutros métodos ou técnicas que lles aportan robustez. Así pois, no segundo capítulo analizaremos diferentes estratexias de globalización que serán útiles para construír métodos globalmente converxentes, é dicir, métodos que converxen, baixo certas condicións, a partir de calquera punto inicial. Estudaremos, en particular, o método híbrido de dicotomía e Newton-Raphson, o método híbrido de dicotomía e regula falsi e, finalmente, falaremos dun método moi robusto que emprega dicotomía, secante e interpolación cadrática inversa: o método de Dekker-Brent.

Abstract

In the first chapter of this dissertation, we introduce the classical bisection, regula falsi, Newton-Raphson, secant and Müller methods for solving non-linear equations in one variable. In practice, none of these methods are presented in isolation because of their problems but are accompanied by other methods or techniques that make robust. We will analyze different globalization strategies that will be useful to construct globally convergent methods, that is, methods that converge, under certain conditions, from any initial point. We will analyze, in particular, the hybrid bisection and Newton-Raphson methods, the hybrid bisection and regula falsi method and finally, we will discuss a very robust method that uses bisection, secant and inverse quadratic interpolation: the Dekker-Brent method.

Introdución

Un dos problemas máis importantes nos diferentes campos da ciencia é o de achar as raíces de ecuacións da forma

$$f(x) = 0 \tag{1}$$

onde $f : \mathbb{R} \rightarrow \mathbb{R}$ é unha función real coñecida. Unha raíz ou solución da ecuación é un valor α que verifica

$$f(\alpha) = 0.$$

Nos problemas onde non é posible atopar a solución analítica, optamos por obter unha aproximación de dita solución mediante métodos numéricos que aproximan de maneira eficiente e con suficiente precisión, as solucións de problemas expresados matematicamente.

No primeiro capítulo deste traballo, introduciremos os métodos clásicos de dicotomía, regula falsi, Newton-Raphson, secante e Müller para resolver (1). Estes métodos preséntanse de maneira illada mentres que na práctica computacional nunca é así xa que non son métodos robustos. Unha vez estudadas as vantaxes e desvantaxes de ditos métodos, en particular, as carencias de converxencia global que presenta, entre outros, o método de Newton-Raphson, centraremonos no segundo capítulo nalgunhas estratexias de globalización para obter métodos con boas propiedades de converxencia (rápidos e globais).

Denotaremos como *método global* ou *algoritmo globalmente converxente* para denotar a un método que está deseñado para converxer a alguna solución de (1) a partir de calquera punto inicial. Presentaremos neste segundo capítulo o método híbrido de dicotomía e Newton-Raphson, o híbrido de dicotomía e regula falsi e finalmente un método híbrido coñecido como o método de Dekker-Brent que emprega secante, interpolación cadrática inversa e dicotomía.

Por último, implementaremos en Matlab, ver anexo A e B, as rutinas dos diferentes métodos numéricos e chegaremos a certas conclusións que nos permitirán validar as vantaxes computacionais que nos aportan os métodos híbridos con respecto aos métodos clásicos.

Capítulo 1

Métodos clásicos

Para entender no capítulo 2 as diferentes estratexias de globalización empregadas para a resolución de ecuacións non lineares nunha variable, introduciremos a continuación os distintos métodos clásicos para resolver a ecuación (1).

1.1. Método de dicotomía

O primeiro método que se presenta, baseado no teorema do valor intermedio, recibe o nome de dicotomía. Suponse que f é unha función continua definida dentro do intervalo $I = [a, b]$ con $f(a)$ e $f(b)$ de signos opostos. O teorema do valor intermedio garante que existe unha raíz α en (a, b) con $f(\alpha) = 0$. A pesar de que o procedemento operará cando haxa máis dunha raíz no intervalo (a, b) , por simplicidade, asumimos que a raíz neste intervalo é única.

Trátase dun algoritmo de *encaixado* porque constrúe unha sucesión de intervalos encaixados $I = I_0 \supset I_1 \supset I_2 \supset I_3 \supset \dots \supset I_n \supset I_{n+1}$ de tal modo que a raíz α que se aproxima pertence a todos eles ($\alpha \in I_n, \forall n \geq 0$) e a sucesión $\{x_n\}_{n \geq 0}$ construída polo algoritmo é tal que $x_n \in I_n \forall n \geq 0$. Dado que a lonxitude dos intervalos $|I_n|$ tende a cero, a converxencia dedúcese inmediatamente sen máis que ter en conta que $|x_n - \alpha| \leq |I_n|$.

O método realiza repetidamente unha redución á metade dos subintervalos de $[a, b]$ e en cada paso localiza a metade que contén a α .

Para comezar, sexa $a_0 = a, b_0 = b$ e x_0 o punto medio de $[a, b]$, ie,

$$x_0 = \frac{a_0 + b_0}{2}.$$

- Se $f(x_0) = 0$ entón $\alpha = x_0$ e rematamos.

- Se $f(x_0) \neq 0$, entón $f(x_0)$ ten o mesmo signo que $f(a_0)$ ou que $f(b_0)$.
 - Se $f(x_0)$ e $f(a_0)$ teñen o mesmo signo entón $\alpha \in (x_0, b_0)$ e consideramos $a_1 = x_0$ e $b_1 = b_0$.
 - Se $f(x_0)$ e $f(a_0)$ teñen signos opostos entón $\alpha \in (a_0, x_0)$ e tomamos $a_1 = a_0$ e $b_1 = x_0$.

Chegamos ó seguinte:

$$[a_1, b_1] \subset [a_0, b_0], \quad b_1 - a_1 = \frac{1}{2}(b_0 - a_0), \quad \text{sign}(f(a_1)) \neq \text{sign}(f(b_1)), \quad \alpha \in (a_1, b_1).$$

En xeral, supoñendo que se constrúe $I_n = [a_n, b_n]$ tense que $\text{sign}(f(a_n)) \neq \text{sign}(f(b_n))$ con $\alpha \in (a_n, b_n)$. Sexa agora

$$x_n = \frac{a_n + b_n}{2},$$

necesariamente ocorre algunha das seguintes posibilidades:

- Se $f(x_n) = 0$ entón $\alpha = x_n$ e rematamos.
- Se $f(x_n) \neq 0$, entón $f(x_n)$ ten o mesmo signo que $f(a_n)$ ou $f(b_n)$.
 - Se $f(x_n)$ e $f(a_n)$ teñen o mesmo signo entón $\alpha \in (x_n, b_n)$ e tomamos $a_{n+1} = x_n$ e $b_{n+1} = b_n$.
 - Se $f(x_n)$ e $f(a_n)$ teñen signos opostos entón $\alpha \in (a_n, x_n)$ e tomamos $a_{n+1} = a_n$ e $b_{n+1} = x_n$.

En calquera caso, cúmprese que:

$$[a_{n+1}, b_{n+1}] \subset [a_n, b_n], \quad b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n), \\ \text{sign}(f(a_{n+1})) \neq \text{sign}(f(b_{n+1})), \quad \alpha \in [a_{n+1}, b_{n+1}].$$

Observación 1.1. Ó implementar o método nunha computadora necesitamos considerar os efectos do erro de redondeo. Por exemplo, o cálculo do punto medio no intervalo $[a_n, b_n]$ debería atopar a partir da ecuación

$$x_n = a_n + \frac{b_n - a_n}{2} \quad \text{en lugar de} \quad x_n = \frac{a_n + b_n}{2},$$

cumpríndose así unha norma xeral de análise numérico que indica que é mellor calcular unha cantidade engadindo unha pequena corrección a unha aproximación previa.

A continuación, detallamos na figura 1.1 a interpretación xeométrica do método de dicotomía. Consideramos unha función f e o intervalo $I_0 = [a_0, b_0]$ que contén á raíz α . Repetimos o proceso iterativo un número finito de veces ata chegar a unha aproximación suficientemente boa da raíz. Dita raíz, α , estará contida no intervalo $I_n = [a_n, b_n]$.

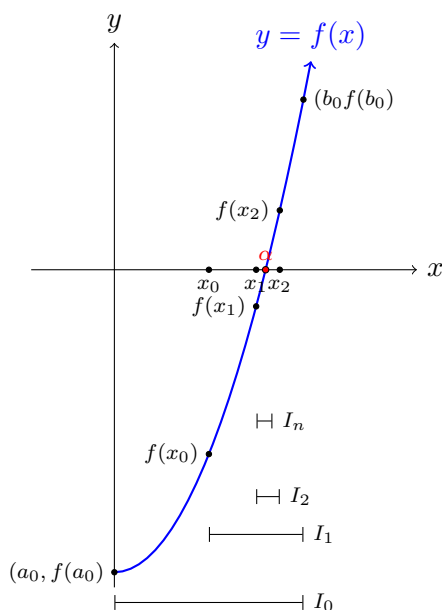


Figura 1.1: Método de dicotomía: interpretación xeométrica

1.1.1. Convergencia e estimación do erro

Teorema 1.2. *Seja $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ unha función continua tal que $f(a)f(b) < 0$ (o que garante a existencia de polo menos unha raíz α no intervalo (a, b)). Entón, a sucesión $\{x_n\}_{n \geq 0}$ xerada polo algoritmo de dicotomía converge a α e ademais cúmprese que:*

$$|\epsilon_n| = |x_n - \alpha| \leq \frac{b - a}{2^{n+1}}.$$

Demostración. En efecto, por construción tense que para calquera $n \geq 0$:

$$a_0 \leq a_1 \leq a_2 \leq \dots \leq a_n < x_n < b_n \leq b_{n-1} \leq \dots \leq b_1 \leq b_0.$$

Desta cadea de desigualdades, dedúcese que a sucesión dos $\{a_n\}_{n \geq 0}$ é monótona crecente pois cada termo é menor ou igual que o termo posterior. Ademais, dita sucesión

está acotada inferiormente (xa que existe un valor a_0 menor ou igual que todos os termos da sucesión) e acotada superiormente (por un valor x_n maior que todos os termos de $\{a_n\}_{n \geq 0}$). É dicir, a sucesión está acotada. Agora, como $\{a_n\}_{n \geq 0}$ é unha sucesión monótona e acotada de números reais entón é converxente. Seguindo un razoamento análogo, deducimos que $\{b_n\}_{n \geq 0}$ é unha sucesión monótona decrecente e acotada. Polo tanto, podemos deducir que $\{b_n\}_{n \geq 0}$ é converxente. En definitiva, ambas sucesións son converxentes.

Sexan agora r e s os seus respectivos límites

$$r = \lim_{n \rightarrow \infty} a_n, \quad s = \lim_{n \rightarrow \infty} b_n.$$

Como

$$b_n - a_n = \frac{b_{n-1} - a_{n-1}}{2} = \frac{b_0 - a_0}{2^n},$$

aplicando límites

$$\lim_{n \rightarrow \infty} (b_n - a_n) = \lim_{n \rightarrow \infty} \frac{b_0 - a_0}{2^n} = \frac{b_0 - a_0}{\infty} = 0$$

e séguese que

$$r = \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = s.$$

Así mesmo, sabemos que $a_n < x_n < b_n$ polo que

$$r = \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} b_n.$$

Ademais, como $f(a_n)f(b_n) < 0$ para todo $n \geq 0$, a continuidade de f asegura que:

$$0 \geq \lim_{n \rightarrow \infty} [f(a_n)f(b_n)] = \lim_{n \rightarrow \infty} f(a_n) \lim_{n \rightarrow \infty} f(b_n) = f(r)f(r) = [f(r)]^2.$$

Polo tanto, necesariamente cúmprese que $f(r) = 0$, pois se $f(r) < 0$ entón $f(r)f(r) > 0$, ou o que é o mesmo, $f(a_n)f(b_n) > 0$ e non estaríamos nas hipóteses do enunciado. Finalmente, como r pertence ao intervalo $[a, b]$ e $f(r) = 0$ entón r é unha raíz (que denotaremos a partir de agora como α) da ecuación $f(x) = 0$ nese intervalo.

Probaremos por último que se verifica a cota do erro sinalada no enunciado. En efecto, para todo $n \geq 0$, cúmprese

$$b_n - a_n = \frac{b - a}{2^n} \quad \text{e} \quad \alpha \in (a_n, b_n).$$

Como $x_n = \frac{a_n + b_n}{2}$ para todo $n \geq 0$, tense por construción que:

$$|x_n - \alpha| \leq \left| \frac{b_n - a_n}{2} \right|.$$

Ademais, como $b_n > a_n$ para todo $n \geq 0$, entón

$$|x_n - \alpha| \leq \left| \frac{b_n - a_n}{2} \right| = \frac{b_n - a_n}{2}.$$

En xeral, comezando para $n = 0$ no intervalo de partida, e mediante un proceso de indución, tras n iteracións, o valor absoluto do erro cometido satisface a desigualdade

$$|x_n - \alpha| \leq \left| \frac{b_n - a_n}{2} \right| = \dots = \frac{b - a}{2^{n+1}}.$$

□

Observación 1.3. Se ao teorema (1.2) lle engadimos a hipótese de que a raíz α é separada no intervalo (a, b) entón a demostración é máis sinxela. En efecto, por construción sabemos que

$$0 \leq |e_n| = |x_n - \alpha| \leq \frac{b - a}{2^{n+1}}.$$

Ademais, por paso ao límite cúmprese que

$$\lim_{n \rightarrow \infty} 0 = 0$$

e

$$\lim_{n \rightarrow \infty} \frac{b - a}{2^{n+1}} = 0,$$

entón, polo teorema de comparación ou tamén chamado teorema do sandwich, podemos concluír que $\lim_{n \rightarrow \infty} |x_n - \alpha| = 0$. Con isto, deducimos que a sucesión xerada polo algoritmo de dicotomía converxe á raíz α en (a, b) .

Observación 1.4. A cota do erro enunciada no teorema 1.2 permite determinar de antemán cantos elementos da sucesión son necesarios construír para garantir unha aproximación a α cun erro menor que ϵ . Deste modo, será suficiente que o número n de iteracións cumpra que:

$$n \geq \frac{\ln\left(\frac{b - a}{\epsilon}\right)}{\ln(2)} - 1. \quad (1.1)$$

Demostración. Basta ter en conta que para n , nesas condicións, cúmprese que

$$|x_n - \alpha| \leq \frac{b-a}{2^{n+1}} < \epsilon$$

ou o que é o mesmo:

$$|x_n - \alpha| \leq \frac{b-a}{\epsilon} < 2^{n+1}.$$

Aplicando logaritmos neperianos a ambos lados da última desigualdade, tense que

$$\ln\left(\frac{b-a}{\epsilon}\right) < \ln(2^{n+1})$$

e agora, por propiedades de logaritmos, resulta que:

$$\ln\left(\frac{b-a}{\epsilon}\right) < (n+1) \cdot \ln(2).$$

Finalmente, operando chegamos a que

$$n \geq \frac{\ln\left(\frac{b-a}{\epsilon}\right)}{\ln(2)} - 1.$$

□

1.1.2. As vantaxes e inconvenientes

Este método ten a propiedade importante de que baixo as hipóteses establecidas, converge sempre a unha solución. Por esta razón, emprégase frecuentemente para pór en marcha métodos máis eficientes. Ademais, é un dos escasos algoritmos onde se pode asegurar que non vai entrar nun bucle infinito debido a que en cada iteración divídese por 2 o tamaño do intervalo $[a, b]$. Polo tanto, ao traballar cunha aritmética de precisión finita nun número finito de iteracións vaise cumprir que $a = b$ e, en consecuencia, o algoritmo parará.

A maior desvantaxe deste método é a súa extremada lentitude. O algoritmo proporciona só unha raíz da ecuación e ningunha información adicional sobre a posible existencia de máis raíces no mesmo intervalo. Ademais, como non se emprega información no referente a forma da curva que representa a propia función f , o algoritmo pode proporcionar aproximacións xa boas sen ter que avanzar ata o final.

1.2. Método de regula falsi

Sexa $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ continua en $[a, b]$ tal que $f(a)f(b) < 0$. Constrúese $\{x_n\}_{n \geq 0}$ da forma:

$$\begin{cases} x_0 = a, & x_1 = b \\ x_{n+1} = \frac{x_m f(x_n) - x_n f(x_m)}{f(x_n) - f(x_m)} = x_n - f(x_n) \frac{x_n - x_m}{f(x_n) - f(x_m)} \end{cases} \quad (1.2)$$

sendo $m = m(n)$ o maior enteiro menor que n tal que $f(x_n)f(x_m) < 0$.

O método de regula falsi consiste en achar a iteración x_{n+1} a partir da abscisa do punto de corte co eixe das x da recta que une os puntos $(x_n, f(x_n))$ e $(x_m, f(x_m))$.

A continuación, amosamos na figura 1.2 a interpretación xeométrica do método de regula falsi. Consideramos unha función f e tomamos $x_0 = a$ e $x_1 = b$, seguidamente achamos os sucesivos iterantes empregando a fórmula iterativa descrita en (1.2).

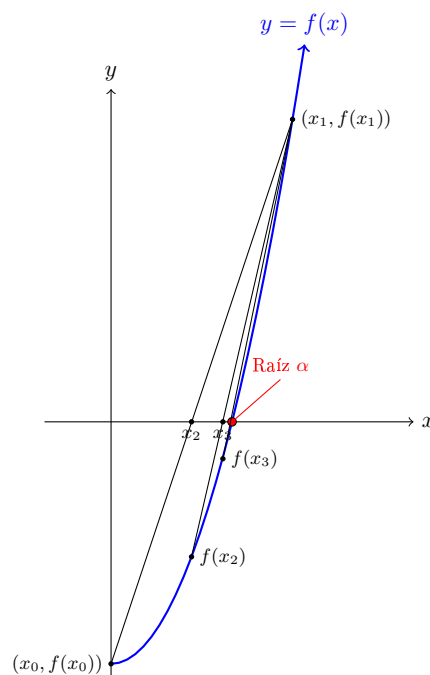


Figura 1.2: Método de regula falsi: interpretación xeométrica

Teorema 1.5 (Converxencia global do método de regula falsi). *Sexa $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ continua con $f(a)f(b) < 0$ e $\alpha \in (a, b)$ unha raíz de f separada en $[a, b]$. A sucesión $\{x_n\}_{n \geq 0}$ construída co método de falsa posición converxe a α .*

Ver a demostración en [9].

1.2.1. As vantaxes e inconvenientes

A converxencia global do método de regula falsi proba que dito método vai converxer sempre baixo certas condicións favorables. Porén, en ocasións, pode presentar situacións de converxencia moi lenta. O seguinte exemplo amosa unha importante desvantaxe do método da falsa posición: a súa unilateralidade. É dicir, conforme se avanza nas iteracións, un dos puntos limitantes do intervalo tende a permanecer fixo. Isto pode levar a unha mala converxencia, especialmente en funcións cunha curvatura importante. Para entender a problemática da que estamos a falar analicemos o seguinte exemplo descrito en [9] e representado na figura 1.3.

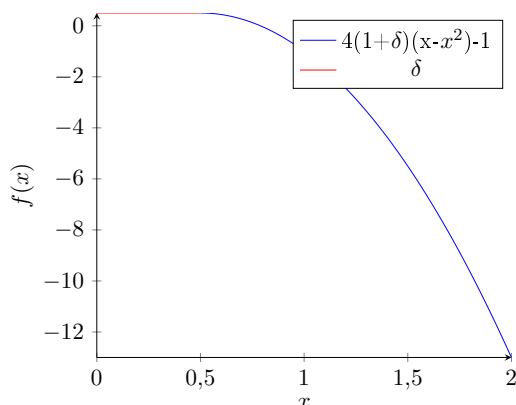


Figura 1.3: Converxencia lenta

Consideremos a función definida por

$$f(x) = \begin{cases} \delta, & \text{se } 0 \leq x \leq \frac{1}{2} \\ 4(1 + \delta)(x - x^2) - 1, & \text{se } \frac{1}{2} \leq x \leq 1 \end{cases}$$

onde $0 < \delta < 1$. Claramente, $f(x)$ e $f'(x)$ son continuas no intervalo $[0, 1]$. Tomamos $x_0 = a = 0$ e $x_1 = b = 1$, deste modo, $f(a) = \delta$ e $f(b) = -1$. Agora, substituíndo na

fórmula do método da falsa posición, temos que

$$x_2 = 1 + 1 \cdot \frac{1 - 0}{-1 - \delta} = 1 - \frac{1}{1 + \delta}.$$

En xeral, mentres que $x_{n-1} \leq \frac{1}{2}$, tense que

$$x_n = 1 - \frac{1}{(1 + \delta)^{n-2}}.$$

Supoñendo que δ é moi pequeno, o número de iteracións para que $x_n \geq \frac{1}{2}$, virá dado por

$$x_n = 1 - \frac{1}{(1 + \delta)^{n-2}} \geq \frac{1}{2},$$

agora aplicando logaritmos neperianos a ambos lados da desigualdade chegamos a que:

$$\ln \left(\frac{1}{(1 + \delta)^{n-2}} \right) \leq \ln \left(\frac{1}{2} \right).$$

Logo,

$$n \geq \frac{\ln(2)}{\ln(1 + \delta)} + 2.$$

Se δ é moi pequeno, o número de iteracións é moi elevado. Por exemplo, se tomamos $\delta = 10^{-6}$ necesítanse arredor de 639.000 iteracións para chegar ao punto 0.5 (que aínda non é a raíz). Pola contra, o método de dicotomía converxe cun erro menor a 2^{-49} en 49 iteracións.

A orixe desta converxencia tan lenta é debido a que as sucesivas iteracións están sempre no mesmo lado da raíz.

Polo xeral, este exemplo ilustra que non é posible realizar xeneralizacións cos métodos de obtención de raíces. Aínda que o método de regula falsi é case sempre superior ao de dicotomía, hai exemplos, coma este, que refutan esta conclusión.

1.3. Método de Newton-Raphson

Supoñamos que $f \in C^2([a, b])$. Se $x_0 \in [a, b]$ é unha aproximación da solución α , de tal forma que $f'(x_0) \neq 0$ e $|\alpha - x_0|$ é pequeno, entón o polinomio de Taylor de primeiro grao centrado en x_0 avaliado en α , exprésase do seguinte xeito:

$$f(\alpha) = f(x_0) + (\alpha - x_0)f'(x_0) + \frac{1}{2}f''(\xi)(\alpha - x_0)^2$$

onde ξ está entre α e x_0 . Como $f(\alpha) = 0$, resulta que

$$0 = f(x_0) + (\alpha - x_0)f'(x_0) + \frac{1}{2}f''(\xi)(\alpha - x_0)^2$$

e truncando o erro, a ecuación queda reducida a

$$0 \approx f(x_0) + (\alpha - x_0)f'(x_0).$$

Despexando agora o valor de α , chegamos a que

$$\alpha \approx x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Isto motiva a construción de sucesións de aproximacións da forma seguinte:

$$\begin{cases} x_0 & \text{dado} \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \end{cases} \quad (1.3)$$

A introdución do método de Newton mediante o desenvolvemento de Taylor resalta a importancia dunha boa aproximación inicial. A suposición crucial é que o termo que contén $(\alpha - x_n)^2$ pode ser eliminado. Esta será unha suposición falsa a menos que x_n sexa unha boa aproximación a α . En particular, se x_0 non está suficientemente preto da raíz, o método de Newton pode non converxer nela.

A continuación, presentamos na figura 1.4 a interpretación xeométrica do método de Newton-Raphson.

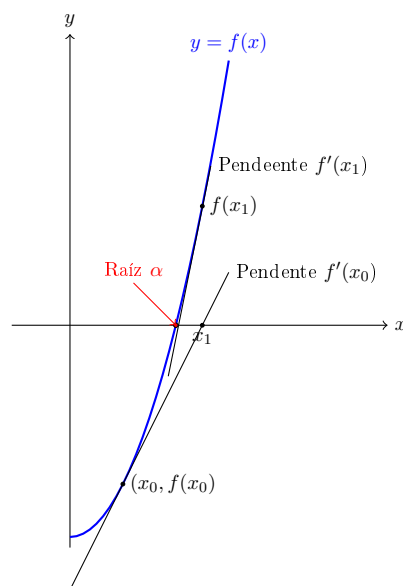


Figura 1.4: Método de Newton-Raphson: interpretación geométrica

1.3.1. Convergencia e erro

Teorema 1.6 (Teorema de convergencia local). *Sea $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ cunha raíz $\alpha \in (a, b)$. Supoñemos f derivable nun entorno $(\alpha - \delta_1, \alpha + \delta_1) \subset [a, b]$, f' continua en α e $f'(\alpha) \neq 0$. Entón:*

1. *A aplicación $g(x) = x - \frac{f(x)}{f'(x)}$ está definida nun entorno $(\alpha - \delta_2, \alpha + \delta_2)$, $\delta_2 < \delta_1$.*
2. *Existe un entorno $(\alpha - \delta_3, \alpha + \delta_3)$ tal que para todo $x_0 \in (\alpha - \delta_3, \alpha + \delta_3)$ a sucesión $\{x_n\}_{n \geq 0}$ con $x_{n+1} = g(x_n)$ converge a α e ademais $x_n \in (\alpha - \delta_3, \alpha + \delta_3)$ para todo $n \geq 0$.*
3. *Para todo $x_0 \in (\alpha - \delta_3, \alpha + \delta_3)$ a convergencia de $\{x_n\}_{n \geq 0}$ é superlineal:*

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|} = 0.$$

A demostración completa pode verse en [9].

De acordo con suposicións razoables, o teorema establece que sempre que se seleccione unha aproximación inicial suficientemente próxima á raíz, o método de Newton converge.

Teorema 1.7 (Converxencia global do método de Newton-Raphson para raíces simples).

Sexa $f : I = [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$, $f \in C^2([a, b])$ cumprindo:

1. $f(a)f(b) < 0$.
2. $f \neq 0$ en $[a, b]$.
3. $f'' \leq 0$ ou $f'' \geq 0$ en $[a, b]$.
4. Se $c \in (a, b)$ denota o extremo de $[a, b]$ no que $|f'|$ é máis pequeno, tense que:

$$\left| \frac{f(c)}{f'(c)} \right| \leq b - a.$$

Entón a ecuación $f(x) = 0$ ten unha única raíz $\alpha \in (a, b)$ e calquera que sexa $x_0 \in [a, b]$, o método de Newton-Raphson converxe a α .

A demostración completa pode verse en [9].

Definición 1.8. (Orde de converxencia) Sexa x_n unha sucesión de números reais tal que

$\lim_{n \rightarrow \infty} x_n = \alpha$ e sexa $p > 0$. Dise que a sucesión ten orde de converxencia p se existe

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^p} = C \in (0, \infty).$$

En calquera caso, a constante C denomínase constante asintótica do erro.

Teorema 1.9 (Orde de converxencia do método de Newton-Raphson). *Supoñamos que α é unha raíz de $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ con $f'(\alpha) \neq 0$ e consideremos ademais que existe un entorno $(\alpha - \delta, \alpha + \delta)$ onde f é dúas veces continuamente derivable. Entón, se $x_0 \neq \alpha$ e $x_n \neq \alpha$ para todo $n \geq n_0$ sendo $n_0 \in \mathbb{N}$, cúmprese que:*

$$\lim_{n \rightarrow \infty} \frac{\epsilon_{n+1}}{\epsilon_n^2} = \frac{f''(\alpha)}{2f'(\alpha)}.$$

En consecuencia, o método é polo menos de orde 2.

Demostración. Da fórmula

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

dedúcese que

$$\epsilon_{n+1} = \epsilon_n - \frac{f(x_n)}{f'(x_n)} = \frac{\epsilon_n f'(x_n) - f(x_n)}{f'(x_n)}.$$

Aplicando agora o teorema de Taylor tense que:

$$0 = f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + \frac{(\alpha - x_n)^2}{2} f''(\xi_n)$$

sendo ξ_n un punto comprendido entre x_n e α . Dedúcese entón que:

$$f(x_n) - \epsilon_n f'(x_n) = -\frac{\epsilon_n^2}{2} f''(\xi_n).$$

e polo tanto

$$\epsilon_{n+1} = \frac{f''(\xi_n)}{2f'(x_n)} \epsilon_n^2.$$

Tendo en conta que nas hipóteses dadas a sucesión converge a α , a continuidade de f' e f'' garante que se $\epsilon_n \neq 0$, ($n \geq n_0$) entón :

$$\lim_{n \rightarrow \infty} \frac{\epsilon_{n+1}}{\epsilon_n^2} = \frac{f''(\alpha)}{2f'(\alpha)}.$$

□

1.3.2. As vantaxes e inconvenientes

Unhas das maiores vantaxes que presenta o método de Newton é a súa velocidade de converxencia con respecto aos outros métodos clásicos. Pola contra, existen situacións onde este método se comporta de maneira deficiente. A continuación, veremos varios exemplos claros.

Como podemos apreciar na figura 1.5, pode ocorrer que escollamos un punto inicial x_0 tal que $f'(x_0) = 0$ ou simplemente que tras aplicar o método de Newton varias veces, cheguemos a obter un punto x_n que teña pendente cero. Isto implica unha división entre cero na fórmula de Newton, o cal significa que a solución dispárase horizontalmente e nunca toca o eixe x.

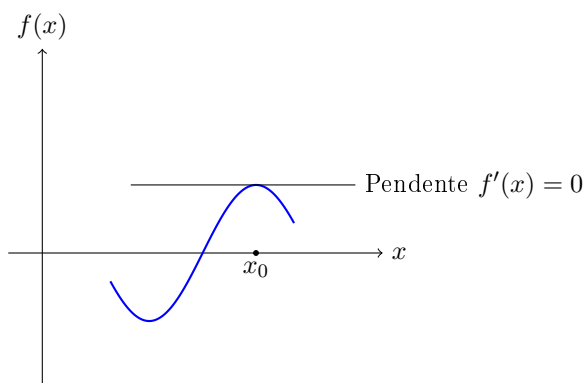


Figura 1.5: Método de Newton-Raphson: problemática 1

A continuación, en la figura 1.6 podemos observar que un valor inicial cercano a la raíz salta a una posición mucho más alejada que la estimación inicial.

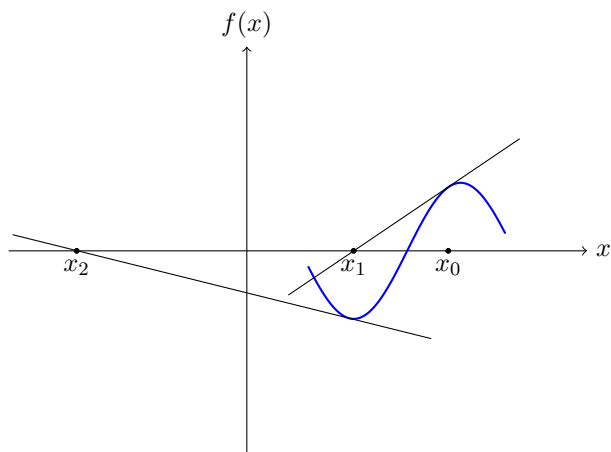


Figura 1.6: Método de Newton-Raphson: problemática 2

Ahora, observemos en la figura 1.7 otra problemática que puede presentarse en Newton: la tendencia a oscilar alrededor de un mínimo o máximo local. Tales oscilaciones pueden persistir o alcanzar una pendiente próxima a cero alejándose tanto de la raíz.

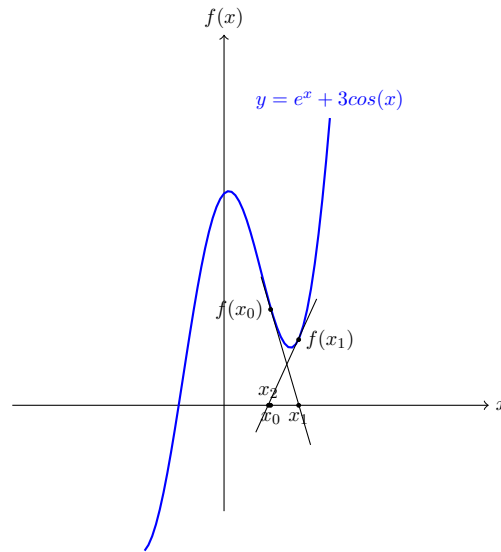


Figura 1.7: Método de Newton-Raphson: problemática 3

Finalmente, na figura 1.8 amosamos o caso onde unha raíz se atopa preto dun punto de inflexión (ie, $f''(x) = 0$). As iteracións que empezan con x_0 diverxen progresivamente da raíz.

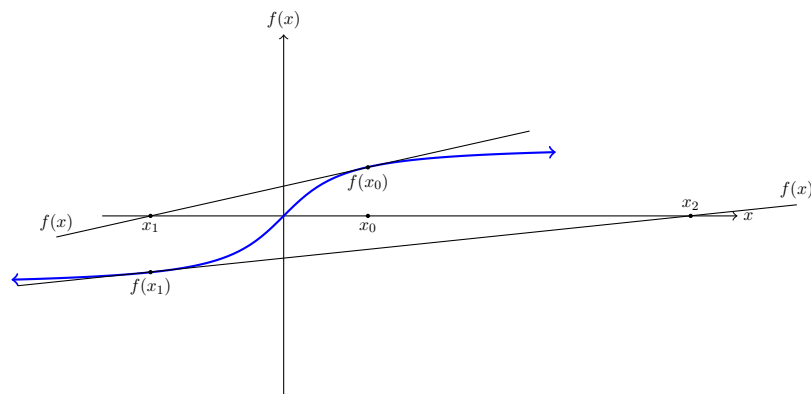


Figura 1.8: Método de Newton-Raphson: problemática 4

Destes exemplos problemáticos deducimos que non hai un criterio xeral de converxencia para o método de Newton-Raphson. A súa converxencia depende da natureza da función e do punto inicial no que se comece. A única solución, nestes casos, é ter un valor inicial que sexa suficientemente próximo á raíz e, porén, para algunhas funcións ningún valor inicial será axeitado. Os bos valores iniciais, xeralmente, predícese cun coñecemento do problema mediante gráficas que proporcionan maior claridade ao comportamento da solución.

Con todo, dito método tamén ten unha debilidade importante que é a necesidade de coñecer o valor da derivada de f en cada aproximación. Para funcións simples e dadas explicitamente, isto non ofrece dificultades serias. Non obstante, se $f'(x)$ é difícil de avaliar ou $f(x)$ é coñecida só implicitamente, resulta impracticable achar $f'(x)$. Polo tanto, pode ser máis conveniente empregar métodos alternativos.

Finalmente, cabe sinalar que poden presentarse situacións de converxencia lenta, especialmente se a elección inicial está lonxe da raíz. En tal caso, o algoritmo comeza cunha fase de busca ata que obtén un valor suficientemente achegado á raíz como para asegurar unha orde cadrática.

1.4. Método da secante

Sexa $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ continua con α raíz de $f(x) = 0$ separada en $[a, b]$. O método da secante está baseado na fórmula de Newton-Raphson, pero evita o cálculo da derivada empregando a seguinte aproximación:

$$f'(x) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

Substituíndo na fórmula de Newton-Raphson, obtemos que

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

Isto induce a construción do seguinte algoritmo:

$$\begin{cases} x_0, x_1 \text{ dado} \\ x_{n+1} = x_n - f(x_n) \frac{(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \end{cases} \quad (1.4)$$

Xeometricamente o valor x_{n+1} é a abscisa do punto de corte co eixe das x da recta que une os puntos $(x_0, f(x_0))$ e $(x_1, f(x_1))$.

A continuación, detallamos na figura 1.9 a interpretación xeométrica do método da secante. Consideramos unha función f , dous puntos iniciais x_0 e x_1 e aplicamos a fórmula iterativa.

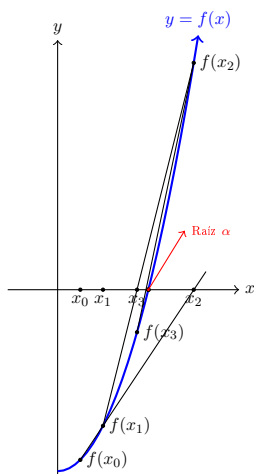


Figura 1.9: Método da secante: interpretação xeométrica

Teorema 1.10 (Orde de converxencia do método da secante). *Supoñamos que α é unha raíz de $f(x) = 0$ que pode obterse como límite da sucesión xerada polo método da secante. Se f é de clase C^2 e $f'(x) \neq 0$, entón a sucesión de iteracións ten orde de converxencia $p = \frac{1 + \sqrt{5}}{2} \approx 1.618$ e*

$$\lim_{n \rightarrow \infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|^p} = \left| \frac{f''(x)}{2f'(x)} \right|^{p-1}$$

Demostración. Da fórmula iterativa do método da secante

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \quad \forall n \geq 1 \quad (1.5)$$

e supoñendo novamente que $x_{n+1} - \alpha = \epsilon_{n+1}$, dedúcese que

$$\epsilon_{n+1} = \epsilon_n - \frac{(\epsilon_n - \epsilon_{n-1})f(\alpha + \epsilon_n)}{f(\alpha + \epsilon_n) - f(\alpha + \epsilon_{n-1})}. \quad (1.6)$$

Agora, empregando a expansión da serie de Taylor para $f(\alpha + \epsilon_{n+1})$ tense que:

$$f(\alpha + \epsilon_{n+1}) \approx f'(\alpha) \left[\epsilon_{n+1} + \frac{\epsilon_{n+1}^2 f''(\alpha)}{2f'(\alpha)} + \dots \right]$$

Calculando de maneira análoga $f(\alpha + \epsilon_n)$ e $f(\alpha + \epsilon_{n-1})$ e substituíndo na ecuación (1.6), cúmprese que

$$\epsilon_{n+1} \approx \epsilon_n - \frac{(\epsilon_n - \epsilon_{n-1})f'(\alpha) \left[\epsilon_n + \frac{\epsilon_n^2 f''(\alpha)}{2f'(\alpha)} + \dots \right]}{f'(\alpha)(\epsilon_n - \epsilon_{n-1}) \left[1 + \frac{1}{2}(\epsilon_n + \epsilon_{n-1}) \frac{f''(\alpha)}{f'(\alpha)} + \dots \right]}$$

ou o que é o mesmo:

$$\begin{aligned} \epsilon_{n+1} &\approx \epsilon_n - \left[\epsilon_n + \frac{\epsilon_n^2 f''(\alpha)}{2f'(\alpha)} + \dots \right] \left[1 + \frac{1}{2}(\epsilon_n + \epsilon_{n-1}) \frac{f''(\alpha)}{f'(\alpha)} + \dots \right]^{-1} = \\ &\epsilon_n - \left[\epsilon_n + \frac{\epsilon_n^2 f''(\alpha)}{2f'(\alpha)} + \dots \right] \left[1 - \frac{1}{2}(\epsilon_n + \epsilon_{n-1}) \frac{f''(\alpha)}{f'(\alpha)} + \dots \right] = \\ &\epsilon_n - \left[\epsilon_n + \frac{\epsilon_n^2 f''(\alpha)}{2f'(\alpha)} - \frac{1}{2}\epsilon_n(\epsilon_n + \epsilon_{n-1}) \frac{f''(\alpha)}{f'(\alpha)} + O(\epsilon_n \epsilon_{n-1}^2) + O(\epsilon_{n-1} \epsilon_n^2) \right]. \end{aligned}$$

Así,

$$\epsilon_{n+1} \approx \epsilon_n - \left[\epsilon_n + \frac{1}{2} \frac{f''(\alpha)}{f'(\alpha)} (\epsilon_n^2 - \epsilon_{n-1}^2 - \epsilon_n \epsilon_{n-1}) \right] = \epsilon_n \epsilon_{n-1} \frac{f''(\alpha)}{2f'(\alpha)}.$$

En particular, obtense a seguinte relación:

$$\epsilon_{n+1} \approx \epsilon_n \epsilon_{n-1} \frac{f''(\alpha)}{2f'(\alpha)} \quad \text{sendo} \quad M = \frac{f''(\alpha)}{2f'(\alpha)}.$$

Agora, calculemos o valor de p :

Supoñamos que dado $p > 0$ e $C > 0$, existe o

$$\lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^p} = C.$$

Por outro lado, como

$$\lim_{n \rightarrow \infty} \frac{\epsilon_{n+1}}{\epsilon_n \epsilon_{n-1}} = M$$

cúmprese que

$$\lim_{n \rightarrow \infty} |\epsilon_{n+1}| = \lim_{n \rightarrow \infty} |\epsilon_n \epsilon_{n-1} M|.$$

En particular,

$$\lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^p} = \lim_{n \rightarrow \infty} \frac{|\epsilon_n \epsilon_{n-1} M|}{|\epsilon_n|^p} = \lim_{n \rightarrow \infty} \frac{|\epsilon_n|^{1-p}}{|\epsilon_{n-1}|^{-1}} |M| = C$$

e ademais verificase que

$$\lim_{n \rightarrow \infty} \frac{|\epsilon_n|^{1-p}}{|\epsilon_{n-1}|^{-1}} = \lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|^{1-p}}{|\epsilon_n|^{-1}} = \frac{C}{|M|}.$$

Polo tanto, temos dous límites que relacionan a ϵ_{n+1} e ϵ_n :

$$\lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|^1}{|\epsilon_n|^p} = C \quad \text{e} \quad \lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|^{1-p}}{|\epsilon_n|^{-1}} = \frac{C}{|M|}.$$

Entón, a única maneira de que o cociente destes termos converxa a un valor finito distinto de cero é que as relacións dos termos dos cocientes dos expoñentes sexan iguais. Isto significa que:

$$\frac{1}{p} = \frac{1-p}{-1}$$

ou o que é o mesmo:

$$p^2 - p - 1 = 0.$$

Resolvendo a ecuación de segundo grao, chegamos a que:

$$p = \frac{1 + \sqrt{5}}{2} \approx 1.618.$$

Finalmente, como xa coñecemos o valor de p , é sinxelo achar o valor de C . Sabemos que

$$\lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^p} = C \tag{1.7}$$

e

$$\lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|^{1-p}}{|\epsilon_n|^{-1}} = \frac{C}{|M|}.$$

Agora, se elevamos a $(1-p)$ o límite (1.7), temos que

$$\lim_{n \rightarrow \infty} \left(\frac{|\epsilon_{n+1}|^1}{|\epsilon_n|^p} \right)^{1-p} = C^{1-p} = \lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|^{1-p}}{|\epsilon_n|^{-1}} = \frac{C}{|M|}.$$

Logo,

$$C^{1-p} = \frac{C}{|M|},$$

é dicir,

$$C = |M|^{\frac{1}{p}} = \left| \frac{f''(x)}{2f'(x)} \right|^{p-1}.$$

□

1.4.1. As vantaxes e inconvenientes

A vantaxe principal deste método respecto ao método de Newton é que non necesita coñecer o valor da derivada. Ademais, presenta unha converxencia rápida (non tanto como o método de Newton) que supera aos métodos de dicotomía e regula falsi.

É un erro moi común pensar que se se comeza con dúas aproximacións iniciais que encerran a raíz, o método da secante producirá iteracións de forma que a raíz está no intervalo de extremos x_n e x_{n-1} . Isto non é así necesariamente. De feito, basta ver a figura 1.9.

Agora ben, cando as aproximacións están moi preto da raíz, poden aparecer problemas de tipo computacional. Son situacións de inestabilidade numérica que hai que evitar na medida do posible. Téñense dado moitas alternativas para o cálculo da seguinte iteración mediante o método da secante. Todas teñen a súas vantaxes e inconvenientes.

A fórmula

$$x_{n+1} = \frac{x_n f(x_{n-1}) - x_{n-1} f(x_n)}{f(x_n) - f(x_{n-1})} \quad (1.8)$$

é moi insatisfactoria dende o punto de vista computacional. A razón é a perda de díxitos significativos que se produce cando as iteracións x_n e x_{n-1} están moi preto da solución.

Un algoritmo para o método da secante baseado na forma usual

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \quad (1.9)$$

ou na forma

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}} \quad (1.10)$$

ten a dificultade dunha moi posible aparición de *overflow*.

As posibilidades de perda de díxitos significativos e *overflow* redúcense no posible escribindo

$$x_{n+1} = x_n - \frac{(x_{n-1} - x_n) \left(\frac{f(x_n)}{f(x_{n-1})} \right)}{1 - \frac{f(x_n)}{f(x_{n-1})}}. \quad (1.11)$$

Supoñendo que

$$|f(x_n)| < |f(x_{n-1})|$$

esta fórmula evita as posibles dificultades e ofrece unha forma exacta de computar x_{n+1} .

A súa vez, o método da secante pode non ser axeitado por alguna das seguintes razóns:

- Pode ocorrer que $f(x_n) = f(x_{n-1})$ de modo que non estaría definido o punto x_{n+1} . Isto apréciase na figura 1.10.
- Tamén pode darse o caso de que algún iterante x_n non pertenza ao dominio de definición de f . Sexa $f = x^2 - 4$ cuxo $Dom f = [0, 3]$. Tomemos como puntos iniciais

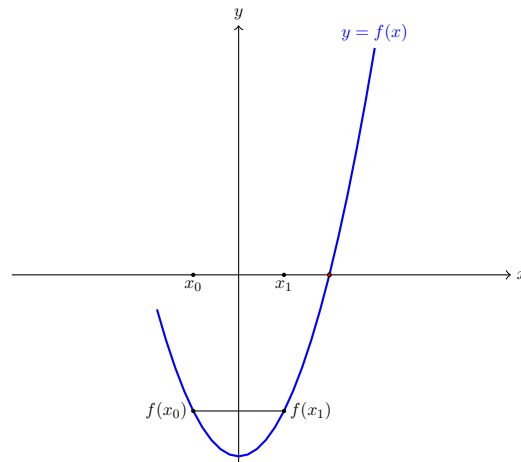


Figura 1.10: Problemática 1: $f(x_0) = f(x_1)$

$x_0 = 0,24$ e $x_1 = 0,59$. Como pode apreciarse na figura 1.11 obtense que o iterante seguinte $x_3 = 5,65$ non pertence ao dominio de definición de f .

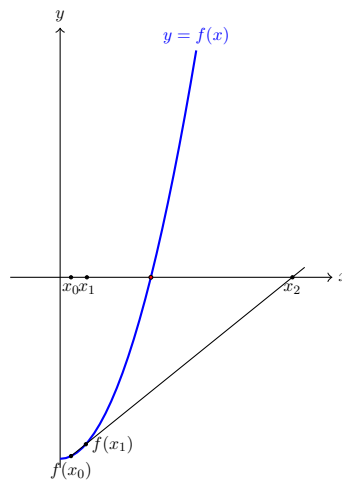


Figura 1.11: Problemática 2: iterante non pertence ao dominio de f .

1.5. Método de Müller

O método de Müller emprega tres aproximacións iniciais dadas: x_0, x_1, x_2 e determina a seguinte aproximación x_3 considerando a intersección do eixe x coa parábola que pasa

polos puntos $(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2))$.

O procedemento de Müller comeza considerando o polinomio cadrático

$$p(x) = a(x - x_2)^2 + b(x - x_2) + c \quad (1.12)$$

que pasa por $(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2))$. Trataremos de atopar valores de a, b, c para que a parábola pase polos puntos indicados, para iso resolvemos o sistema de ecuacións:

$$p(x_0) = a(x_0 - x_2)^2 + b(x_0 - x_2) + c = f(x_0) \quad (1.13)$$

$$p(x_1) = a(x_1 - x_2)^2 + b(x_1 - x_2) + c = f(x_1) \quad (1.14)$$

$$p(x_2) = c = f(x_2) \quad (1.15)$$

Deducimos da última ecuación que:

$$c = f(x_2).$$

Polo tanto, substituíndo este valor nas dúas primeiras ecuacións chegamos a que:

$$a(x_0 - x_2)^2 + b(x_0 - x_2) = f(x_0) - f(x_2) \quad (1.16)$$

$$a(x_1 - x_2)^2 + b(x_1 - x_2) = f(x_1) - f(x_2) \quad (1.17)$$

Para resolver este sistema imos definir as diferenzas que nel aparecen. Nomearemos:

$$h_0 = x_1 - x_0 \quad (1.18)$$

$$h_1 = x_2 - x_1 \quad (1.19)$$

$$\delta_0 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (1.20)$$

$$\delta_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (1.21)$$

Desta forma, se manipulamos as ecuacións anteriores temos:

$$\left. \begin{aligned} (h_0 + h_1)b - (h_0 + h_1)^2 a &= h_0 \delta_0 + h_1 \delta_1 \\ h_1 b - h_1^2 a &= h_1 \delta_1 \end{aligned} \right\} \quad (1.22)$$

Se resolvemos o sistema anterior chegamos a:

$$\left. \begin{aligned} a &= \frac{\delta_1 - \delta_0}{h_1 - h_0} \\ b &= ah_1 + \delta_1 \\ c &= f(x_2) \end{aligned} \right\} \quad (1.23)$$

Unha vez achados os coeficientes do polinomio de segundo grao, temos definida a parábola

$$y = a(x - x_2)^2 + b(x - x_2) + c.$$

A continuación, para atopar a aproximación da raíz temos que facer $y = 0$ na expresión anterior. Das dúas solucións posibles, tomamos aquela que minimize a expresión:

$$|x_3 - x_2|.$$

A nova aproximación da raíz, x_3 , virá dada pola solución de $p(x_3) = 0$. Polo tanto:

$$a(x_3 - x_2)^2 + b(x_3 - x_2) + c = 0.$$

De aquí, temos que

$$x_3 - x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2ac} \quad (1.24)$$

Nas ecuacións de segundo grao onde $b^2 \gg 4ac$ a diferenza no numerador pode ser moi pequena. Neste caso, podemos escribir a diferenza $x_3 - x_2$ do seguinte xeito:

$$x_3 - x_2 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}} \quad (1.25)$$

Se despexamos x_3 , tense que:

$$x_3 = x_2 + \frac{-2c}{b \pm \sqrt{b^2 - 4ac}} \quad (1.26)$$

Tomaremos agora o signo de forma que a diferenza $x_3 - x_2$ sexa o menor posible en valor absoluto. Para iso, consideraremos o signo de modo que coincida co signo de b , desta forma o denominador será máis grande e polo tanto a raíz será máis próxima a x_2 .

Así:

$$x_3 = x_2 - \frac{2c}{b + \text{signo}(b)\sqrt{b^2 - 4ac}} \quad (1.27)$$

Onde a, b, c están dadas polas expresións anteriores. Unha vez se determina x_3 , o procedemento reiníciase usando x_1, x_2, x_3 en lugar de x_0, x_1, x_2 para determinar a seguinte aproximación x_4 . O método continúa ata que se obtén unha conclusión satisfactoria. Xa que o método involucra en cada paso ó radical $\sqrt{b^2 - 4ac}$, o método aproximará raíces complexas cando sexa apropiado. En xeral,

$$x_{n+1} = x_n - \frac{2c_n}{b_n + \text{signo}(b_n)\sqrt{b_n^2 - 4a_n c_n}} \quad (1.28)$$

A continuación, vexamos na seguinte figura 1.12 a interpretación xeométrica do método de Müller.

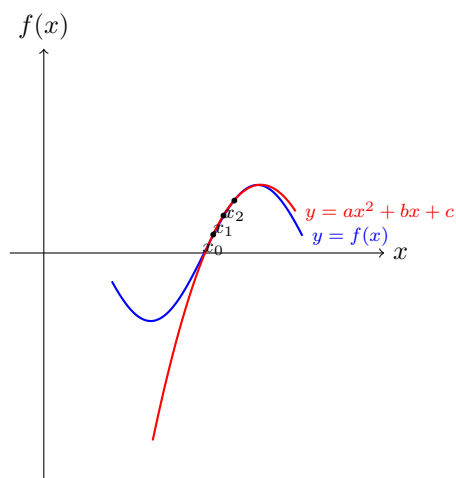


Figura 1.12: Método de Müller: interpretación xeométrica

1.5.1. As vantaxes e inconvenientes

A vantaxe principal que presenta este método é que permite obter tanto raíces complexas como raíces reais. O método de Müller está baseado no método da secante pero emprega unha aproximación cadrática en vez dunha aproximación lineal. Isto implica unha converxencia máis rápida que o método da secante.

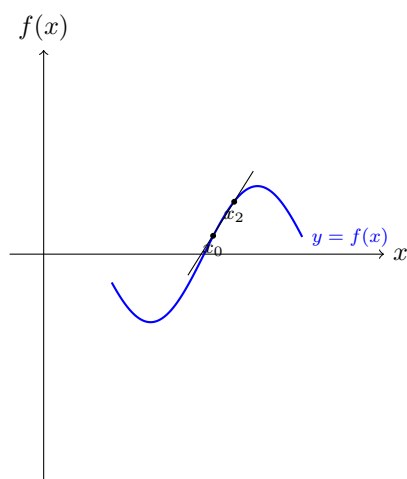


Figura 1.13: Método da secante

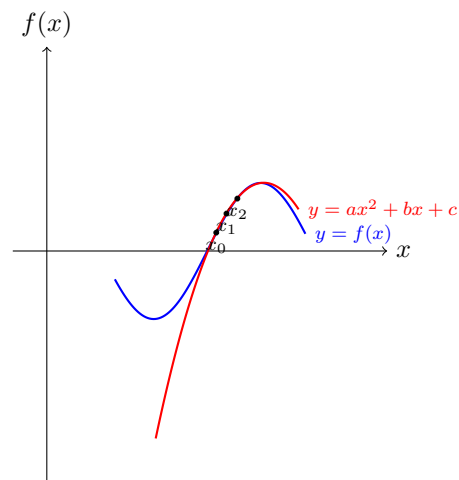


Figura 1.14: Método de Müller

Así mesmo, unha deficiencia a sinalar neste método é a seguinte: supoñamos que para algún $n \geq 0$ cúmprese que $f(x_n) = f(x_{n+1}) = f(x_{n+2}) \neq 0$, dese xeito a ecuación cadrática redúcese a unha función constante distinta de cero e nunca interseca ó eixe x . Vexámolo na figura 1.15.

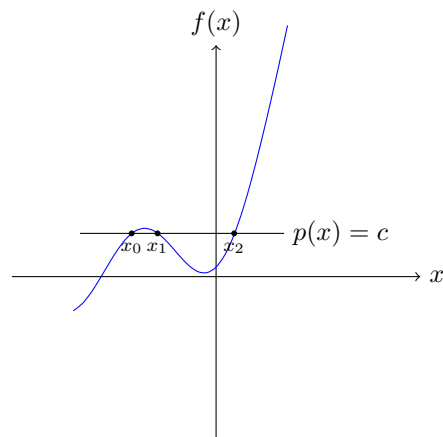


Figura 1.15: Método de Müller: problemática

Capítulo 2

Estratexias de globalización do método de Newton-Raphson

No capítulo 1 introducimos os métodos clásicos para resolver ecuacións non lineares nunha variable e ademais presentamos ditos métodos de maneira illada coas súas vantaxes e os seus inconvenientes. Tal como vimos, o método de Newton preséntase rápido, mais carente de converxencia global. En consecuencia, na práctica computacional os métodos clásicos nunca se empregan de forma illada, senón acompañados doutros métodos ou técnicas que lles agregan robustez. Neste capítulo, presentaremos métodos híbridos elementais pero robustos. En particular, estudaremos o método híbrido de dicotomía e Newton-Raphson e o método híbrido de dicotomía e regula falsi. Para rematar, presentaremos o método híbrido de Dekker-Brent que emprega secante, interpolación cadrática inversa e dicotomía.

2.1. Método híbrido de Dicotomía e Newton-Raphson

O obxectivo desta sección consiste en crear un método robusto e eficiente que se caracterice por aproveitar as boas propiedades do método de dicotomía e Newton-Raphson. Un primeiro algoritmo que podemos establecer consiste en aplicar o método de dicotomía ata achar un intervalo suficientemente pequeno. A finalidade é acadar a rexión onde o método de Newton ten converxencia local e a partir de aí empregar dito algoritmo. Con todo, a rexión onde converxe localmente é descoñecida e, en consecuencia, non podemos predicir cantas iteracións do método de dicotomía serán suficientes. Por unha banda, poucas iteracións poden provocar que o algoritmo de Newton non converxa. Por outra banda, demasiadas iteracións iniciais poden causar que a converxencia sexa moi lenta e non se aproveite a converxencia cadrática do método de Newton. Polo tanto, aínda que se trata dunha forma interesante de proceder, vexamos que existe outra máis acertada.

Un segundo algoritmo a establecer é o seguinte:

1. Calcular o punto medio do intervalo inicial $[a, b]$.
2. Empregar Newton sempre que sexa posible, noutras palabras, sempre que a derivada da función inicial avaliada no punto sexa distinta de cero e que o punto que se ache esté dentro do intervalo (a, b) . Noutro caso, retrocedemos e aplicamos dicotomía.

A continuación, presentamos o pseudocódigo do método híbrido e seguidamente, na figura 2.1, o diagrama de fluxo:

Entrada: $f, a, b, tol, imax, delta$.

Saída: Raíz x , n (n^o iteracións), $a, b, f(a), f(b), f(x)$.

$x = (a + b)/2$

mentres $abs(f(x)) \geq tol$ e $i < imax$ **facer**

se $df(x) > delta$ **entón**

$x = x - f(x)/df(x)$ (Newton-Raphson)

se non

se $f(a)f(x) < 0$ **entón**

$b = x$

se non

$a = x$

fin se

$x = (a + b)/2$ (dicotomía)

fin se

se $x < a$ e $x > b$ **entón**

 actualizar o intervalo

$x = (a + b)/2$ (dicotomía)

fin se

$i = i + 1$

fin mentres

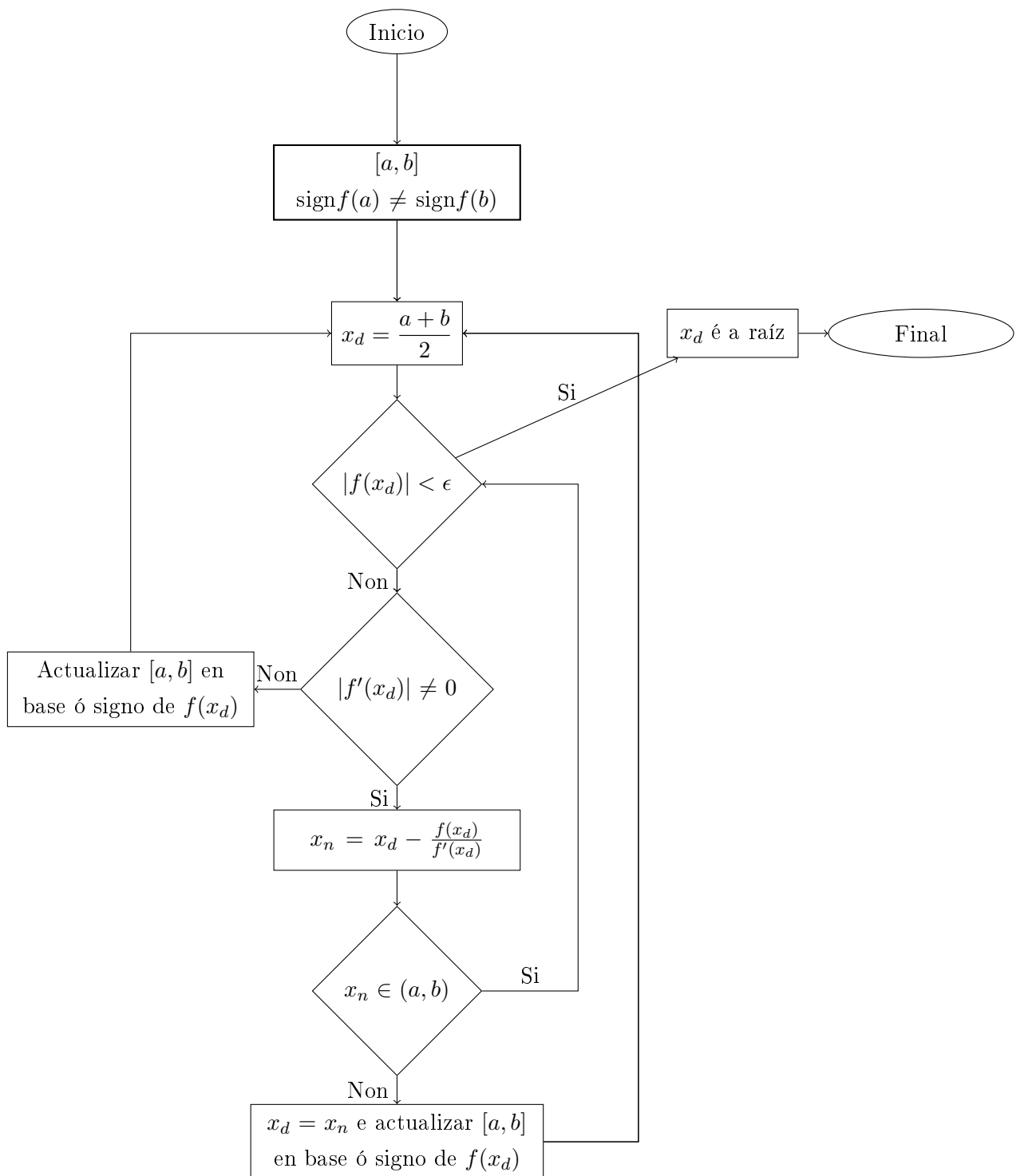


Figura 2.1: Diagrama de fluxo do método híbrido.

2.1.1. Resultados

Exporemos agora unha serie de resultados obtidos na práctica computacional que nos permitirán contrastar a información que estamos a dar. Facendo uso dos programas implementados en Matlab no Anexo A e B, consideremos a ecuación

$$f(x) = x^4 + 3x^3 - 15x^2 - 2x + 9 = 0$$

e tomemos como $\text{tol} = 10^{-6}$, $\text{imax} = 100$ (o número máximo de iteracións) e a, b os extremos do intervalo de estudo. Nas seguintes táboas móstranse os resultados obtidos ao aplicar o método de Newton-Raphson, o método de dicotomía e o método híbrido de dicotomía e Newton-Raphson.

Observamos que os primeiros elementos das sucesións do método de Newton coinciden cos primeiros elementos das sucesións construídas con dicotomía e tamén das sucesións construídas co híbrido. Destacar que todos estes métodos converxen e ademais, o máis rápido na súa converxencia é método de Newton. Pola contra, como podemos apreciar, as diferenzas no número de iteracións para chegar a unha boa aproximación por parte do método híbrido e de Newton son mínimas.

Método de Newton-Raphson								
	[-6,-4], $x_0 = -5$		[-2,0], $x_0 = -1$		[0,2], $x_0 = 1$		[2,4], $x_0 = 3$	
n	x	f(x)	x	f(x)	x	f(x)	x	f(x)
1	-5.834646	-73.066818	-0.818182	0.599959	0.789474	-4.000000	2.690722	5.878503
2	-5.602754	7.102661	-0.795438	-0.022743	0.785975	-0.0633513	2.593333	0.486799
3	-5.574890	0.095019	-0.795067	-0.000372	0.785972	-0.00004	2.583692	0.004514
4	-5.574507	0.000018	-0.795067	0.000000	0.785972	0.000000	2.583601	0.000000
5	-5.574506	0.000000						

Método de dicotomía								
	[-6,-4]		[-2,0]		[0,2]		[2,4]	
n	x	f(x)	x	f(x)	x	f(x)	x	f(x)
1	-5.000000	-106.000000	-1.000000	-6.000000	1.000000	-4.000000	3.000000	30.000000
2	-5.500000	-17.812500	-0.500000	5.937500	0.500000	4.687500	2.500000	-3.812500
3	-5.750000	47.363281	-0.750000	1.113281	0.750000	0.644531	2.750000	9.644531
4	-5.625000	12.834229	-0.875000	-2.157959	0.875000	-1.638428	2.625000	2.135010
5	-5.562500	-2.960190	-0.812500	-0.450668	0.812500	-0.482407	2.562500	-1.024155
6	-5.593750	4.817491	-0.781250	0.349244	0.781250	0.085267	2.593750	0.507859
7	-5.578125	0.898990	-0.796875	-0.046229	0.796875	-0.197589	2.578125	-0.269887
8	-5.570313	-1.037988	-0.789063	0.152628	0.789063	-0.055907	2.585938	0.116032
9	-5.574219	-0.071349	-0.792969	0.053480	0.785156	0.014745	2.582031	-0.077663
10	-5.576172	0.413357	-0.794922	0.003695	0.787109	-0.020565	2.583984	0.019000
11	-5.575195	0.170888	-0.795898	-0.021250	0.786133	-0.002906	2.583008	-0.029378
12	-5.574707	0.049741	-0.795410	-0.008773	0.785645	0.005920	2.583496	-0.005200
13	-5.574463	-0.010812	-0.795166	-0.002538	0.785889	0.001507	2.583740	0.006897
14	-5.574585	0.019463	-0.795044	0.000579	0.786011	-0.000699	2.583618	0.000848
15	-5.574524	0.004325	-0.795105	-0.000979	0.785950	0.000404	2.583557	-0.002176
16	-5.574493	-0.003243	-0.795074	-0.000200	0.785980	-0.000148	2.583588	-0.000664
17	-5.574509	0.000541	-0.795059	0.000190	0.785965	0.000128	2.583603	0.000092
18	-5.574501	-0.001351	-0.795067	-0.000005	0.785973	-0.000010	2.583595	-0.000286
19	-5.574505	-0.000405	-0.795063	0.000092	0.785969	0.000059	2.583599	-0.000097
20	-5.574507	0.000068	-0.795065	0.000043	0.785971	0.000025	2.583601	-0.000003
21	-5.574506	-0.000169	-0.795066	0.000019	0.785972	0.000008	2.583602	0.000044

Método híbrido de dicotomía e Newton-Raphson								
	[-6,-4]		[-2,0]		[0,2]		[2,4]	
n	x	f(x)	x	f(x)	x	f(x)	x	f(x)
1	-5.000000	-106.000000	-0.8181818	-6.0000000	1.000000	-4.0000000	3.000000	30.000000
2	-5.834646	73.066818	-0.7954384	-0.5999590	0.789474	-0.0633513	2.690722	5.878503
3	-5.602754	7.102661	-0.7950667	-0.0094933	0.785975	-0.0000511	2.593333	0.486799
4	-5.574890	0.095019	-0.7950666	-0.0000025			2.583692	0.004514
5	-5.574507	0.000018	-0.7950666	0.0000000				

Agora ben, supoñamos que a elección do punto inicial para o método de Newton-Raphson non é tan acertada. Consideremos o intervalo $[1.06, 2.7]$ e o punto de arranque $x_0 = 1.88$. Na seguinte táboa móstrase que o método de dicotomía necesita realizar 21 iteracións para achar unha boa aproximación da raíz.

Método de dicotomía		
n	x	f(X)
1	1.880000	-15.350001
2	2.290000	-10.713948
3	2.495000	-4.020127
4	2.597500	0.698046
5	2.546250	-1.784014
6	2.571875	-0.574394
7	2.584688	0.053889
8	2.578281	-0.262226
9	2.581484	-0.104663
10	2.583086	-0.025511
11	2.583887	0.014158
12	2.583486	-0.005684
13	2.583687	0.004235
14	2.583586	-0.000725
15	2.583636	0.001755
16	2.583611	0.000515
17	2.583599	-0.000105
18	2.583605	0.000205
19	2.583602	0.000050
20	2.583601	-0.000027
21	2.583601	0.000011

Pola contra, cando empregamos o método de Newton, deducimos que non hai converxencia xa que claramente $f'(x_0) \approx 0$. Agora ben, empregando o método híbrido, amosado na seguinte táboa, deducimos que fan falta unicamente 5 iteracións para atopar unha boa aproximación da raíz.

Método híbrido de dicotomía e Newton-Raphson		
n	x	f(x)
1	1.880000	-15.350001
2	2.290000	-10.713948
3	2.508359	-3.460353
4	2.589770	0.307484
5	2.583638	0.001822

2.1.2. Conclusións

O algoritmo híbrido de dicotomía e Newton-Raphson é unha estratexia de globalización que permite achar a raíz da ecuación $f(x) = 0$ nun curto número de iteracións. Ademais de competir en velocidade con respecto aos métodos clásicos de dicotomía e Newton-Raphson, é capaz de solventar (como se observa na derradeira táboa) as problemáticas que atopabamos no método de Newton. Polo tanto, concluimos que estamos ante un algoritmo robusto e eficaz.

2.2. Método híbrido: dicotomía e regula falsi

A continuación, a seguinte estratexia de globalización que podemos establecer consiste en empregar dúas características:

- A redución sucesiva á metade do intervalo $[a, b]$ grazas ao método de dicotomía.
- A converxencia global, baixo condición razoables, tanto do método de regula falsi como do método de dicotomía.

Estas dúas propiedades permiten construír un método robusto que asegura, baixo certas condicións, que o algoritmo sempre converxe. Como veremos, dita converxencia é veloz. Ironicamente, o método de dicotomía é o encargado de evitar a posibilidade de que o algoritmo híbrido quede varado nunha converxencia lenta.

O procedemento a seguir consiste en empregar selectivamente o algoritmo que mellor se adecúe entre dicotomía e falsa posición para redefinir sucesivamente tanto a raíz aproximada como o intervalo que a contén.

O primeiro paso neste método será achar dous puntos:

1. O punto medio, m , entre $[a, b]$.
2. O punto s que é a abscisa de intersección co eixe x da recta que une os puntos $(a, f(a))$ e $(b, f(b))$ tal que $f(a)f(b) < 0$.

A continuación, o segundo paso é calcular o valor absoluto da función aplicada ao punto medio, m , e contrastar se dito valor é menor ou maior que o valor absoluto da función aplicada ao punto secante s . No caso de que o valor absoluto sexa menor, ie, que m esté máis preto da raíz, o método a seguir é dicotomía. Noutro caso, empregamos o método de

regula falsi. O proceso repítese as veces precisas ata chegar a cumprir un criterio de parada axeitado ou un número máximo de iteracións.

Seguidamente, presentamos o pseudocódigo do método híbrido de dicotomía e regula falsi:

Entrada: $f, a, b, tol, imax$.

Saída: Raíz x e o intervalo que a contén.

mentres $i < imax$ **facer**

$$m = \frac{a+b}{2}$$

$$Errm = |f(m)|$$

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

$$Errs = |f(s)|$$

se $Errm < Errs$ **entón**

$$x = m$$

$$Erra = Errm$$

se $f(a)f(x) < 0$ **entón**

$$b = x$$

se non

$$a = x$$

fin se

se non

$$x = s$$

$$Erra = Errs$$

se $f(a)f(x) < 0$ **entón**

$$b = x$$

se non

$$a = x$$

fin se

$$i = i + 1$$

fin se

fin mentres

2.2.1. Resultados

Unha vez explicado o procedemento teórico desta estratexia de globalización, imos presentar un exemplo práctico programado en Matlab que permite corroborar a información que estamos a dar. Para elo, consideremos a ecuación $f = x^4 - x^3 - 1 = 0$ e supoñamos que $\text{tol} = 10^{-6}$, $\text{imax} = 100$, $a = -0,7$ e $b = 1,5$. Presentamos os resultados correspondentes nas seguintes táboas:

Método de dicotomía		
[-0.7,1.5]		
n	x	f(x)
1	0.400000	-1.038400
2	0.950000	-1.042869
3	1.225000	-0.586390
4	1.362500	-0.083109
5	1.431250	0.264374
6	1.396875	0.081749
7	1.379688	-0.002832
8	1.388281	0.038912
9	1.383984	0.017905
10	1.381836	0.007503
11	1.380762	0.002327
12	1.380225	-0.000254
13	1.380493	0.001036
14	1.380359	0.000391
15	1.380292	0.000068
16	1.380258	-0.000093
17	1.380275	-0.000013
18	1.380283	0.000028 0.000062
19	1.380279	0.000008
20	1.380277	-0.000002
21	1.380278	0.000003
22	1.380278	0.000000

Método de regula falsi		
[-0.7,1.5]		
n	x	f(x)
1	0.130478	-1.001931
2	0.942685	-1.048014
3	1.279227	-0.415478
4	1.362390	-0.083612
5	1.377311	-0.014186
6	1.379791	-0.002335
7	1.380198	-0.000382
8	1.380265	-0.000063
9	1.380275	-0.000010
10	1.380277	-0.000002
11	1.380278	0.000000

Método híbrido de dicotomía e regula falsi						
[-0.7,1.5]						
n	a	x	b	f(a)	f(x)	f(b)
1	-0.700000	0.130478	1.500000	-0.416900	-1.001931	0.687500
2	0.130478	0.942685	1.500000	-1.001931	-1.048014	0.687500
3	0.942685	1.279227	1.500000	-1.048014	-0.415478	0.687500
4	1.279227	1.378722	1.389614	-0.415478	-0.007453	0.045481
5	1.378722	1.380256	1.389614	-0.007453	-0.000105	0.045481
6	1.380256	1.380277	1.389614	-0.000105	-0.000001	0.045481
7	1.380277	1.380278	1.389614	-0.000001	-0.000000	0.045481

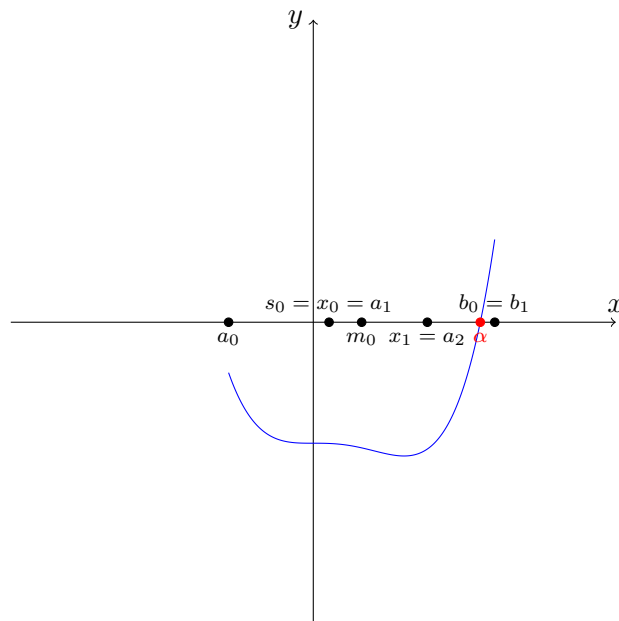


Figura 2.2: Método híbrido: dicotomía e regula falsi

A continuación, especificamos a construcción dos dous primeiros iterantes do método híbrido para facilitar o seu entendemento. Realizando os cálculos pertinentes, temos que

$$m_0 = 0.4, \quad s_0 = 0.130478$$

e ademais, satisfácese que

$$1.001931 = |f(s_0)| < |f(m_0)| = 1.038400$$

polo que deducimos que a mellor aproximación á raíz é $x_0 = s_0$. A continuación, actualizamos o intervalo $[a, b]$.

$$f(a)f(x_0) = f(-0.7)f(0.130478) = -0.4169 \cdot (-1.001931) = 0.417705 > 0$$

entón, necesariamente

$$a_1 = x_0 \text{ e } b_0 = b_1.$$

En particular,

$$a_1 = 0.130478 \text{ e } b_1 = 1.5$$

Na segunda iteración, calculamos o iterante x_1 a partir de $m_1 = 0.815239$ e $s_1 = 0.9427$. Como $1.1 = |-1.1| = |f(m)| > |f(s)| = |-1.048| = 1,0480$ tomamos como raíz o punto $s_1 = x_1 = 0.9427$. Ademais, ao cumprirse que $f(a_1)f(x_1) > 0$, temos necesariamente que $a_2 = s_1 = x_1$ e $b_2 = b_1$. Repetimos o proceso un número finito de veces ata achar unha boa aproximación da raíz da ecuación.

2.2.2. Conclusións

O método híbrido de dicotomía e regula falsi consegue solventar a problemática da unilateralidade que, en ocasións, presenta o método illado da falsa posición. Deste modo, evítase que o método quede fixado nun punto do intervalo e ademais, mellora o número de iteracións con respecto aos métodos clásicos de dicotomía e regula falsi.

2.3. Método de Dekker-Brent

O método de Dekker-Brent é un método híbrido que combina dicotomía, secante e interpolación cadrática inversa con algunhas características adicionais que o fan moi robusto e xeralmente moi eficiente. Os métodos de dicotomía e secante foron analizados xa no capítulo 1. Nesta sección examinaremos brevemente o método de interpolación cadrática inversa.

2.3.1. Interpolación cadrática inversa

O método de interpolación cadrática inversa é unha técnica que se emprega para resolver a ecuación $f(x) = 0$, porén, este algoritmo rara vez se emprega por si só. Neste caso, mostráremolo formando parte do método de Dekker-Brent. Dita técnica consiste en aproximar a función f por unha parábola que pasa por 3 puntos consecutivos $(x_i, f(x_i))$ sendo $i = \{n, n - 1, n - 2\}$.

A continuación, consideramos o seguinte exemplo para xeneralizar dito método. Sexa unha aplicación $y = f(x)$, en particular, sexa $f(x) = x^3 - x^2 - x - 1$ e tomemos os puntos $x_0 = 0$, $x_1 = 1$ e $x_2 = 2$. Empregando o polinomio de interpolación de Lagrange:

$$p(x) = f(x_1) \cdot \frac{x - x_2}{x_1 - x_2} \cdot \frac{x - x_3}{x_1 - x_3} + f(x_2) \cdot \frac{x - x_1}{x_2 - x_1} \cdot \frac{x - x_3}{x_2 - x_3} + f(x_3) \cdot \frac{x - x_1}{x_3 - x_1} \cdot \frac{x - x_2}{x_3 - x_2}$$

e substituíndo os valores de x_0, x_1, x_2 en $p(x)$, resulta que

$$p(x) = 2x^2 - 3x - 1.$$

A súa vez, podemos calcular mediante f as imaxes dos puntos x_0, x_1 e x_2 . Logo, $y_0 = f(x_0) = -1$, $y_1 = f(x_1) = -2$ e $y_2 = f(x_2) = 1$. Agora, asumindo que f ten unha función cadrática inversa g , dedúcese que $x_0 = g(-1)$, $x_1 = g(-2)$ e $x_2 = g(-3)$.

O obxectivo é achar a parábola que pasa por eses tres puntos dados e tomar a intersección da parábola co eixe x como a nova estimación da raíz. Así,

$$g(y) = x_1 \cdot \frac{y - y_2}{y_1 - y_2} \cdot \frac{y - y_3}{y_1 - y_3} + x_2 \cdot \frac{y - y_1}{y_2 - y_1} \cdot \frac{y - y_3}{y_2 - y_3} + x_3 \cdot \frac{y - y_1}{y_3 - y_1} \cdot \frac{y - y_2}{y_3 - y_2}.$$

Substituíndo agora os valores y_0, y_1, y_2 en $g(y)$, chegamos a que:

$$g(y) = \frac{2y^2}{3} + y + \frac{1}{3}.$$

Finalmente, considerando $y = 0$, dedúcese que

$$g(0) = \frac{1}{3}.$$

Este valor induce a estimación da nova raíz: $x_3 = \frac{1}{3}$.

Agora, xeralizando para $n \geq 2$, temos que:

$$g(y) = x_{n-2} \cdot \frac{y - y_{n-1}}{y_{n-2} - y_{n-1}} \cdot \frac{y - y_n}{y_{n-2} - y_n} + x_{n-1} \cdot \frac{y - y_{n-2}}{y_{n-1} - y_{n-2}} \cdot \frac{y - y_n}{y_{n-1} - y_n} + x_n \cdot \frac{y - y_{n-2}}{y_n - y_{n-2}} \cdot \frac{y - y_{n-1}}{y_n - y_{n-1}}$$

e considerando $y = 0$, resulta que:

$$x_{n+1} = x_{n-2} \cdot \frac{0 - y_{n-1}}{y_{n-2} - y_{n-1}} \cdot \frac{0 - y_n}{y_{n-2} - y_n} + x_{n-1} \cdot \frac{0 - y_{n-2}}{y_{n-1} - y_{n-2}} \cdot \frac{0 - y_n}{y_{n-1} - y_n} + x_n \cdot \frac{0 - y_{n-2}}{y_n - y_{n-2}} \cdot \frac{0 - y_{n-1}}{y_n - y_{n-1}}.$$

En definitiva, chegamos ó seguinte método iterativo:

$$\begin{aligned}
 & x_n, x_{n-1}, x_{n-2} \quad \text{dados para todo } n \geq 2. \\
 x_{n+1} = & x_{n-2} \cdot \frac{y_{n-1}y_n}{(y_{n-2} - y_{n-1})(y_{n-2} - y_n)} + x_{n-1} \cdot \frac{y_{n-2}y_n}{(y_{n-1} - y_{n-2})(y_{n-1} - y_n)} + \\
 & + x_n \cdot \frac{y_{n-2}y_{n-1}}{(y_n - y_{n-2})(y_n - y_{n-1})}
 \end{aligned}$$

2.3.2. Algoritmo de Dekker-Brent

O método de Dekker-Brent é a base da rutina *fzero* en Matlab. A idea xeral para a procura da raíz é, sempre que sexa posible, empregar secante ou interpolación cadrática inversa. No caso de que isto xere un resultado inaceptable, é dicir, unha estimación da raíz que esté demasiado preto dun punto final do intervalo ou demasiado preto a un punto inicial do intervalo, o algoritmo volve ao método de dicotomía. Aínda que o método de dicotomía pode ser máis lento, xera unha estimación da raíz con garantía de estar dentro da rexión de converxencia debido á súa converxencia global.

Comezamos considerando unha función f , o extremo inferior a do intervalo $[a, b]$, o extremo superior do intervalo $[a, b]$, unha tolerancia $tol = 10^{-6}$ e $\delta = 4\epsilon|b| + tol$ sendo ϵ o épsilon da máquina. Ao inicio, elíxese un terceiro punto c coincidente con a . Cando o algoritmo progresa, en cada paso calcúlase un conxunto de tres novos puntos a, b, c , onde:

- b é a nova aproximación da raíz.
- a é o valor previo de b .
- c é o punto máis recente para o cal $f(b)f(c) < 0$.

De forma que b e c sempre encerran á raíz.

Máis precisamente, en cada paso estudamos se $|f(b)| < |f(a)|$ e en caso contrario intercambiamos a por b e b por a . A continuación, se $f(a) \neq f(c)$ e $f(b) \neq f(c)$ entón calculamos s mediante a técnica de interpolación cadrática inversa.

$$s = a \frac{f(b)f(c)}{(f(a) - f(b))(f(a) - f(c))} + b \frac{f(a)f(c)}{(f(b) - f(a))(f(b) - f(c))} + c \frac{f(a)f(b)}{(f(c) - f(a))(f(c) - f(b))}.$$

En caso contrario, optamos por empregar o método da secante

$$s = b - f(b) \frac{b - a}{f(b) - f(a)}.$$

Agora ben, Brent fai adicións moi precisas para evitar que o algoritmo converxa de forma lenta. Se se cumpre algunha das seguintes condicións, ie,

1. Se $s \notin [\frac{3a+b}{4}, b]$ ou
2. $|s - b| \geq \frac{|b - c|}{2}$ ou
3. $|s - b| \geq \frac{|c - d|}{2}$ ou
4. $|b - c| < |\delta|$ ou
5. $|c - d| < |\delta|$

entón empregamos o método de dicotomía e polo tanto recalculamos s como segue:

$$s = \frac{a + b}{2}.$$

Finalmente, consideramos $d = c$, $c = b$ e actualizamos o intervalo $[a, b]$. Se $f(a)f(s) < 0$ entón $b = s$ e senón $a = s$. Repetimos o proceso de busca da raíz ata que se cumpra un dos seguintes criterios de parada $abs(f(b)) < tol$ ou $|b - a| < tol$ ou $f(s) < tol$.

Devolvemos o valor de b que é a nova raíz.

Expoñemos un exemplo práctico para facilitar o seu entendemento.

Supoñamos que $f = x^3 + x^2 - 5x + 3$ e sexa o intervalo $[a_0, b_0] = [-4, \frac{4}{3}] = [4, 1.333333]$, e $\delta = 2\epsilon b_n$ $tol = 10^{-6}$. Se avaliamos en f os extremos a_0 e b_0 , resulta que $f(a_0) = -25$ e $f(b_0) = 0.481481$. Ademais, como se cumpre que $f(a_0)f(b_0) < 0$, entón podemos afirmar a existencia de polo menos unha raíz no intervalo $[a_0, b_0]$.

A continuación, comprobamos que se satisface a condición $|f(b_0)| < |f(a_0)|$. En efecto, $0.481481 < 25$. Agora (soamente na primeira iteración) tomamos $c_0 = a_0$, é dicir, $c_0 = -4$ (nas restantes iteracións, suponse que $c_n = b_n$ para todo $n \geq 1$). Como $f(c_0) = f(a_0)$ debemos empregar o método da secante:

$$s_0 = b_0 - f(b_0) \frac{b_0 - a_0}{f(b_0) - f(a_0)} = \frac{4}{3} - f\left(\frac{4}{3}\right) \frac{\frac{4}{3} + 4}{0.481481 + 25} = 1.232558.$$

Pola contra, este valor s_0 aínda non é decisivo. Como ben vimos, no caso de cumprirse unha ou máis condicións establecidas no algoritmo, calcularíamos s_0 empregando dicotomía e rexeitaríamos o valor de s_0 anteriormente achado.

O seguinte paso é realizar as comprobacións pertinentes. a primeira é ver se

$$s \notin \left[\frac{3a_0 + b_0}{4}, b_0 \right].$$

É claro que isto non é certo xa que $1.232558 = s \in [-2.666666, 1.333333]$. A segunda comprobación será ver se se cumpre a seguinte desigualdade:

$$|s_0 - b_0| \geq \frac{|b_0 - c_0|}{2}.$$

Dita condición non se cumpre pois

$$0.100775 \not\geq 2.666666.$$

Pola contra, como d_n só está definido para $n \geq 1$, non estudamos a terceira e quinta condición expresada no algoritmo xa que estas relacionan o valor de d_0 . Finalmente, vexamos se satisface a cuarta:

$$|b_0 - c_0| < |\delta|.$$

É claro que non se cumpre xa que:

$$|1.333333 + 4| = 5.333333 \not< |4 \cdot \epsilon \cdot \frac{4}{3} + tol|.$$

En conclusión, como ningunha condición se satisface, concluímos que o mellor método para achar s_0 é utilizar secante.

A continuación, supoñamos $d_1 = c_0$ en particular, $d_1 = -4$ e $c_1 = b_0$, ie, $c_1 = 1.333333$. Como $f(a_0)f(s_0) = -5,722766 < 0$, entón necesariamente $b_0 = s_0$ e, polo tanto, o novo intervalo de estudo será $[a_1, b_1] = [-4, 1.232558]$. Finalmente, como $|b_0 - a_0| \not< tol$ ou $\text{abs}(f(s_0)) \not< tol$ ou $\text{abs}(f(b_0)) \not< tol$ entón necesariamente temos que seguir iterando.

Vexamos agora dunha forma máis breve o procedemento a seguir na segunda iteración. Neste caso, $a_1 = -4$, $f(a_1) = -25$, $b_1 = 1.232558$ e $f(b_1) = 0.228910$. Logo, $f(a_1)f(b_1) < 0$ e podemos asegurar que existe polo menos unha raíz nese intervalo. Ademais, compróbase que se cumpre a condición $|f(b_1)| < |f(a_1)|$. Agora, como $c_1 = 1.333333$ e $f(c_1) = 0.481481$ entón deducimos que $f(c_1) \neq f(a_1)$ e $f(b_1) \neq f(c_1)$ e estamos en condicións de empregar

o método de interpolación cadrática inversa. Operando, resulta que $s_1 = 1.142052$.

Comprobemos se o cálculo do valor s mediante este método é o acertado, ou se pola contra, deberíamos empregar dicotomía. Pois ben, con unhas sinxelas operacións, próbase que non se cumpre ningunha das 5 condicións impostas no algoritmo, daquela, a técnica a seguir será interpolación cadrática inversa.

Sexa $d_2 = c_1 = 1.333333$ e $c_2 = b_1 = 1.232558$. Como $f(a_1)f(s_1) < 0$ entón $a_2 = -4$ e $b_2 = 1.142052$. Debemos seguir iterando posto que nin $|b_1 - a_1| \not\leq tol$ nin $abs(f(s_1)) \not\leq tol$ nin $abs(f(b)) \not\leq tol$.

Para concluír, realicemos unha terceira iteración no intervalo $[a_2, b_2] = [-4, 1.142052]$. É claro que $f(a_2)f(b_2) < 0$, polo que podemos asegurar a existencia de polo menos unha raíz nese intervalo. Ademais, cúmprese que $|f(b_2)| < |f(a_2)|$, $c_2 = 1.232558$ e $f(c_1) = 0.228910$. Como $f(c_2) \neq f(a_2)$ e $f(b_2) \neq f(c_2)$, deste xeito empregamos o método de interpolación cadrática inversa e $s = 1.090317$. Porén, nesta ocasión vemos que non é o mellor método para proceder xa que se satisface a seguinte condición:

$$\frac{|b_2 - c_2|}{2} \leq |s_2 - b_2|.$$

Pois, $0.045253 \leq 0.0517$, así que debemos retroceder na nosa busca e aplicar o método de dicotomía

$$s_2 = \frac{b_2 + a_2}{2} = -1.428974.$$

Agora, como $f(s_2)f(a_2) < 0$, entón necesariamente $a_3 = -4$ e $b_3 = 1.428974$. Por ser $f(s_2) \not\leq tol$ ou $f(b_2) \not\leq tol$, $|b_2 - a_2| \not\leq tol$ seguimos iterando un número finito de veces ata que se cumpra un dos tres criterios de parada establecidos. Despois de 7 iteracións máis, chegamos a que a raíz aproximada é -3.000000 . A continuación, amosamos o pseudocódigo do método de Dekker-Brent.

Entrada: f, a, b, tol .

Saída: raíz b .

mentres $abs(f(b)) > 0$ ou $abs(f(s)) > 0$ ou $|b - a| > tol$ **facer**

se $|f(a)| < |f(b)|$ **entón**

intercambiamos (a, b)

fin se

$c = a$

se $f(a) \neq f(c)$ e $f(b) \neq f(c)$ **entón**

$$s = a \frac{f(b)f(c)}{(f(a) - f(b))(f(a) - f(c))} + b \frac{f(a)f(c)}{(f(b) - f(a))(f(b) - f(c))} + c \frac{f(a)f(b)}{(f(c) - f(a))(f(c) - f(b))} \quad (\text{interpolación cadrática inversa})$$

se non

$$s = b - f(b) \frac{b - a}{f(b) - f(a)} \quad (\text{método da secante})$$

fin se

se $s \notin \left[\frac{3a + b}{4}, b \right]$ ou $|s - b| \geq \frac{|b - c|}{2}$ ou $|s - b| \geq \frac{|c - d|}{2}$ ou $|b - c| < |\delta|$ ou $|c - d| < |\delta|$ **entón**

$$s = \frac{a + b}{2} \quad (\text{método de dicotomía})$$

fin se

$d = c$

$c = b$

se $f(a)f(s) < 0$ **entón**

$b = s$

se non

$a = s$

fin se

fin mentres

2.3.3. Conclusións

O método de Dekker-Brent trátase dun dos algoritmos para a resolución de ecuacións non lineares nunha variable máis robustos e eficientes ata agora coñecidos. O emprego do método de dicotomía consegue resolver a problemática da mala converxencia que, en ocasións, pode presentar o método da secante e interpolación cadrática inversa. Pola contra, estes dous últimos métodos conseguen evitar que o algoritmo de Dekker-Brent poida converxer dunha forma moi lenta.

Apéndices

Apéndice A

Código Matlab para os métodos clásicos

A.1. Método de dicotomía

```
1 clear
2 clc
3 syms x
4 Fun=x^4+3*x^3-15*x^2-2*x+9; a=0;b=2;tol=10^-6;imax=100;
5 f=inline(Fun);
6 i=1;
7 A=-4; B=4;
8 fprintf(' i          a          x          b          f(a)
          f(x)          f(b)          Err\n');
9 while i<imax
10     x=a+(b-a)/2;
11     X(i)=x;
12     Err=(b-a)/2;
13     fprintf('%4d %11.6f %11.6f %11.6f %11.6f %11.6f %11.6f %11.6f\n',
14             ,i,a,x,b,f(a),f(x),f(b),Err)
15         if abs(f(x))<tol || Err<tol
16             fprintf('\nA solución aproximada e x=%11.8f\n',x);
17             break;
18         end
19     if f(a)*f(x)>0
```

```
19         a=x;
20     else
21         b=x;
22     end
23     i=i+1;
24 end
25 %Elaboracion da grafica
26 z=A:0.01:B;
27 figure(gcf)
28 plot(z,f(z),'r')
29 hold on
30 plot(X,f(X),'+b')
31 text(X(end),f(X(end)),'←raiz')
32 grid on
33 hold off
```

A.2. Método de regula falsi

```

1 clear
2 clc
3 syms x
4 Fun=x^4-x^3-1; a=-0.7; b=1.5; tol=10^-6; imax=100;
5 f=inline(Fun);
6 i=1;
7 fprintf(' i          a          x          b          f(a)
          f(x)          f(b)\n');
8 while i<imax
9     x=a-f(a)*(b-a)/(f(b)-f(a));
10    X(i)=x;
11    Err=abs(x-a);
12    fprintf(' %4d %11.6f %11.6f %11.6f %11.6f %11.6f %11.6f\n',i,a,x
          ,b,f(a),f(x),f(b));
13    if Err<tol || abs(f(x))<tol
14        fprintf('\nA solucion aproximada e x=%11.8f\n:',x);
15        break;
16    else
17        if f(x)*f(a)<0
18            b=x;
19        else
20            a=x;
21        end
22    i=i+1;
23    end
24 end
25 %Elaboracion da grafica
26 z=x-2:0.01:x+2;
27 plot(z,f(z),'r')
28 hold on
29 plot(X,f(X),'+b')
30 text(X(end),f(X(end)),'<—raiz')
31 grid on
32 hold off

```

A.3. Método de Newton-Raphson

```

1 clear
2 clc
3 format long
4 syms x
5 Fun=x^4+3*x^3-15*x^2-2*x+9;x0=1.88; a=0;b=2;tol=10^-6;imax=100;
6 f=inline(Fun);
7 Df=diff(Fun);
8 df=inline(Df);
9 i=1;
10 fprintf(' i           p           Eabs           Erel           |f(x)|\n');
11 while i<imax
12     x=x0-f(x0)/df(x0);
13     X(i)=x;
14     Eabs=abs(x-x0);
15     Erel=abs(Eabs/x);
16     En=abs(f(x));
17     fprintf('%4d %11.6f %11.6f %11.6f %11.6f\n',i,x,Eabs,Erel,abs
18             (f(x)))
19     if or (Erel<tol,Eabs<tol)
20         fprintf('La aproximacion a la raiz es: %12.9f\n',x')
21         break;
22     else
23         x0=x;
24     end
25     i=i+1;
26 end
27 %Elaboracion da grafica
28 z=-4:0.01:4;
29 plot(z,f(z),'r')
30 hold on
31 plot(X,f(X),'+;b')
32 text(X(end),f(X(end)),'<—raiz')
33 grid on
34 hold off

```

A.4. Método da secante

```

1 syms x
2 format long
3 Fun=exp(-x)+cos(x); x0=-2; x1=2; tol=10^-5; imax=100;
4 f=inline(Fun);
5 i=1;
6 y0=f(x0);
7 y1=f(x1);
8 fprintf('      n      x0      x      x1      f(x0)
          |f(x)|      f(x1)      Errabs\n');
9 while i<imax
10     x=x1-y1*(x1-x0)/(y1-y0);
11     X(i)=x;
12     Errabs=abs(p-p1);
13     fprintf('%4d %11.6f %11.6f %11.6f %11.6f %11.6f %11.6f %11.6f\n
            ',i,x0,x,x1,f(x0),f(x1),Errabs,abs(f(x)));
14     if Errabs<tol || abs(f(x))<tol
15         fprintf(' O valor da aproximacio e: % 12.9f\n ',x)
16         break;
17     else
18         x0=x1;
19         y0=y1;
20         x1=x;
21         y1=f(x);
22     end
23     i=i+1;
24 end
25 %Elaboracion da grafica
26 z=x-2:0.01:x+2;
27 plot(z,f(z),'r')
28 hold on
29 plot(X,f(X),'+;b')
30 text(X(end),f(X(end)),'<—raiz')
31 grid on
32 hold off

```

A.5. Método de Müller

```

1  clc
2  clear
3  syms x
4  x0=-1;x1=0;x2=1;tol=10^-5;imax=100;Fun=exp(-x)+cos(x);
5  f=inline(Fun);
6  i=1;
7  h1=x1-x0;
8  h2=x2-x1;
9  delta1=(f(x1)-f(x0))/h1;
10 delta2=(f(x2)-f(x1))/h2;
11 d=(delta2-delta1)/(h2+h1);
12 i=3;
13 fprintf('    i          x0          x1          x2          x
           f(x)\n');
14
15 while i<imax
16     b=delta2+h2*d;
17     D=sqrt((b^2-4*f(x2)*d));
18     if abs(b-D)<abs(b+D)
19         E=b+D;
20     else
21         E=b-D;
22     end
23     h=-2*f(x2)/E;
24     x=x2+h;
25     if abs(h)<tol
26         fprintf('\nA solución aproximada e x=%11.8f\n:',x);
27         break;
28     end
29 fprintf('%4d %11.6f %11.6f %11.6f %11.6f %11.6f\n',i,x0,x1,x2,x,f
        (x));
30 x0=x1;
31 x1=x2;
32 x2=x;

```

```
33 h1=x1-x0 ;  
34 h2=x2-x1 ;  
35 delta1=(f(x1)-f(x0))/h1 ;  
36 delta2=(f(x2)-f(x1))/h2 ;  
37 d=(delta2-delta1)/(h2+h1) ;  
38 i=i+1 ;  
39 end
```


Apéndice B

Código Matlab para os métodos híbridos

B.1. Método híbrido de dicotomía e Newton-Raphson

```
1 function HDN
2 syms x
3 Fun=exp(-x)+cos(x); a=-2; b=2; tol=10^-5; imax=100; delta=10^-5
4 A=a;B=b;
5 f=inline(Fun);
6 Df=diff(Fun);
7 df=inline(Df);
8     if f(a)*f(b)<0
9         x=(a+b)/2;
10    i=1;
11    fprintf('   i           a           x           b           f(a)
12           f(x)           f(b)\n');
13 while fx >=tol && i<imax
14     X(i)=x;
15     fprintf(' %d %11.6f %11.6f %11.6f %11.6f %11.6f %11.6f\n',i,a,x,b
16           ,f(a),f(x),f(b));
17     if df(x)< delta
18         x=x-f(x)/df(x);
19     else
20         [a,b]=biseccion(a,b,x,Fun);
```

```
19         x=(a+b) /2;
20     end
21     if a<=x && x>=b
22         [a,b]= biseccion (a ,b ,x ,Fun) ;
23         x=(a+b) /2;
24     end
25     i=i+1;
26 end
27 end
28
29 function [a ,b]= biseccion (a ,b ,x ,Fun)
30 f=inline (Fun) ;
31 if f(a)*f(x)>0
32     a=x;
33 else
34     b=x;
35
36 end
37 %Elaboracion da grafica
38 z=A:0.01:B;
39 figure ( gcf)
40 plot (z , f (z) , 'r')
41 hold on
42 plot (X, f (X) , '+b')
43 grid on
44 text (X(end) , f (X(end)) , '<—raiz')
45 hold off
46 end
```

B.2. Método híbrido de dicotomía e regula falsi

```

1  clc
2  clear
3  syms x
4  format long;
5  Fun=x^4-x^3-1; a=-0.7; b=1.5; tol=10^-6; imax=100;
6  f=inline(Fun);
7  fprintf('      n      a      x      b      f(a)
      f(x)      f(b)\n');
8  while i<imax
9      m=(a+b)/2;
10     Errm=abs(f(m));
11     s=a-f(a)*(b-a)/(f(b)-f(a));
12     Errs=abs(f(s));
13     if Errm< Errs %o metodo de dicotomia determina o novo
        intervalo e a raiz
14         x=m;
15         if abs(f(x))<tol
16             fprintf('\nA solucion aproximada e p=%11.8f\n:',x);
17             break;
18         end
19         Erra=Errm;
20         if f(a)*f(x)>0
21             a=x;
22         else
23             b=x;
24         end
25     else %emplegamos o metodo de falsa posicion
26         x=s;
27         %X(i)=x;
28         fprintf('%4d %11.6f %11.6f %11.6f %11.6f %11.6f %11.6f\n',i,a,x,
                b,f(a),(f(x)),(f(b)));
29         Erra=Errs;
30         if abs(f(x))<tol
31             fprintf('\nA solucion aproximada e p=%11.8f\n:',x);

```

```
32         break ;
33     end
34     if f(a)*f(x)<0
35         b=x;
36     else
37         a=x;
38     end
39 end
40 i=i+1;
41 end
42
43 %Elaboracion da grafica
44 z=-p:0.01:p;
45 plot(z,f(z),'r')
46 hold on
47 plot(P,f(P),'+;b')
48 text(P(end),f(P(end)),'←raiz')
49 grid on
50 hold off
```

B.3. Método de Dekker-Brent

```

1 clear
2 clc
3 syms x
4 f(x)=x^3+x^2-5*x+3;
5 a=-4;
6 b=4/3;
7 tol=10^(-6);
8 if f(a)*f(b)>0
9 %disp('Non existe raiz no intervalo [a,b]')
10 return;
11 else
12 %disp('Existe raiz en tal intervalo [a,b]')
13 end
14
15 if abs(f(b))<=abs(f(a))
16 %disp('Cumprese a condicion desexada')
17 else
18     temp=a;
19     a=b;
20     b=temp;
21 end
22     c=a;
23     d=b;
24     N=100;
25     i=1;
26     fprintf(' i      a      s      b      f(a)
                f(s)      f(b)      c      d      (3a+b)/4\n'
                )
27 while i<=N || abs(b-a)<tol || abs(f(s))<tol || abs(f(b))<tol
28
29 if f(a)~=f(c)    &&    f(b)~=f(c)
30
31 s=double(a*f(b)*f(c)/((f(a)-f(b))*(f(a)-f(c)))+b*f(a)*f(c)/((f(b)
    )-f(a))*(f(b)-f(c))+c*f(a)*f(b)/((f(c)-f(a))*(f(c)-f(b))));

```

```
32 else
33
34 s=double(b-f(b)*(b-a)/(f(b)-f(a)));
35     end
36     if (s<(3*a+b)/4 || s>b) || abs(s-b)>abs(b-c)/2
37         || abs(s-b)>=abs(c-d)/2 || (abs(b-c)<tol)
38         || (abs(c-d)<tol)
39
40 s=(a+b)/2;
41     end
42     d=c;
43     c=b;
44
45 fprintf(' %3d %10.4f %10.4f %10.4f %10.4f %10.4f %10.4f %10.4f %10.4f
46 %10.4f\n',i,a,s,b,f(a),f(s),f(b),c,d,(3*a+b)/4)
47     if f(a)*f(s)<0
48         b=s;
49     else
50         a=s;
51     end
52
53     if abs(f(a))<abs(f(b))
54         temp=a;
55         a=b;
56         b=temp;
57     end
58
59 i=i+1;
60 end
```

Bibliografía

- [1] Altaee, A. A. H., Hoomod, H. K. e Hussein, K.A. (2015). *A New Approach to Find Roots of Nonlinear Equations by Hybrid Algorithm to Bisection and Newton-Raphson Algorithms*. Iraqi journal for Information Technology, 7, 75-82.
- [2] Benítez, F. e Díaz, J. M. (1998). *Introducción a los métodos numéricos para la resolución de ecuaciones*. Servicio de publicaciones de la universidad de Cádiz.
- [3] Brent, R. P. (2013). *Algorithms for Minimization without Derivatives*. Courier Corporation.
- [4] Burden, R. L. e Faires, J. D. (1985). *Análisis Numérico*. Grupo Editorial Iberoamérica, México.
- [5] Chapra, S. C. e Canale, R. P. (2007). *Métodos numéricos para ingenieros*. McGraw-Hill, México.
- [6] Fernández, D. R e Pilotta, E. (2004). *El método de Newton y estrategias de globalización para la resolución numérica de ecuaciones no lineales*. Revista de Educación Matemática, 19(1), 3–19.
- [7] Sabharwal, C. L. (2019). *Blended Root Finding Algorithm Outperforms Bisection and Regula Falsi Algorithms*. Mathematics, 7(11), 1118.
- [8] Sabharwal, C. L. (2021). *An Iterative Hybrid Algorithm for Roots of Non-Linear Equations*. Eng, 2(1), 80–98.
- [9] Viaño, J. M. e Burguera, M. (1997). Lecciones de Métodos Numéricos 2. *Resolución de ecuaciones numéricas*. Tórculo.