DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Máster Universitario en Computación Paralela y Distribuida
Algoritmos Paralelos Matriciales en Ingeniería

Tema 6.

Resolución de ecuaciones no lineales

Bibliografía

- **C.T.Kelley**
"Iterative Methods for Linear and Nonlinear Equations" SIAM, 1995

- J.E.Dennis, R.Schnabel
"Numerical Methods for Uncostrained Optimization and Nonlinear Equations". Prentice-Hall Series in Computational Mathematics. Cleve Moler Advisor. 1983

- J.Wilkinson
"The Algrebraic Eigenvalue Problem". Oxford Univ. Press. 1965

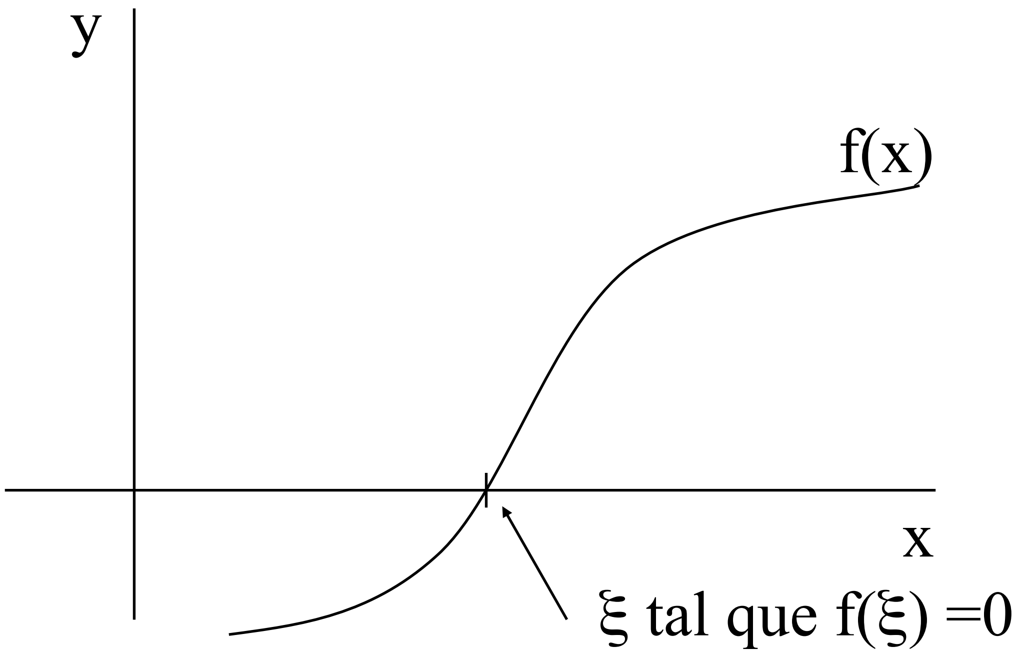
Estructura Prevista de la Parte 3:

Tema 6. Resolución de Ecuaciones No lineales

(previo al problema simétrico de valores propios)

Tema 7. Resolución de Sistemas Generales No Lineales. Optimización

Resolución de ecuaciones



Objetivo: Encontrar la raíz ξ de $f(x)$, si esta existe.

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Descripción del problema:

Hallar soluciones de la ecuación $f(x)=0$

Ejemplo 1: ecuación de primer grado

$$ax + b = 0$$

despejamos

Ejemplo 2: ecuación de segundo grado;

$$ax^2 + bx + c = 0$$

despejamos usando la fórmula $\dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Descripción del problema

Ejemplo 3: ecuación de quinto grado;

$$ax^5 + bx^4 + cx^3 + dx + e = 0$$

Ejemplo 4: ecuación con funciones no algebraicas

$$\cos(e^x + x) + x = 0$$

Para aquellos problemas para los que no se conozca método analítico para resolver f(x)=0, utilizaremos **Métodos Iterativos.**

Es posible que no exista solución: cos(x)-3=0

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

IMPORTANTE

¿Valdrán los métodos para ecuaciones que no sean de la forma $f(x)=0$; Por ejemplo, $\cos(x)=\text{sen}(x)-0.2$?

RESPUESTA:

Se resta: $\cos(x)-\text{sen}(x)+0.2=0$

y a continuación, se aplica alguno de los métodos que se van a estudiar

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Métodos Numéricos

Generamos sucesión: $x_0, x_1, x_2, \ldots, x_n$ $\longrightarrow$ $\boxed{\lim_{n\to\infty} x_n = \xi}$

$\downarrow$

Los métodos numéricos para hallar raíces de funciones serán iterativos

$\downarrow$

Se parará cuando $|f(x_n)| < \text{tolerancia}$,
o cuando $|x_{n+1} - x_n| < \text{tolerancia}$
o cuando $|x_{n+1} - x_n| / |x_{n+1}| < \text{tolerancia}$

Razón de Convergencia

Si existen dos constantes K y α y un entero N tales que:

$$|x_{n+1} - \xi| \approx K\,|x_n - \xi|^{\alpha} \quad \text{con} \quad n \geq N$$

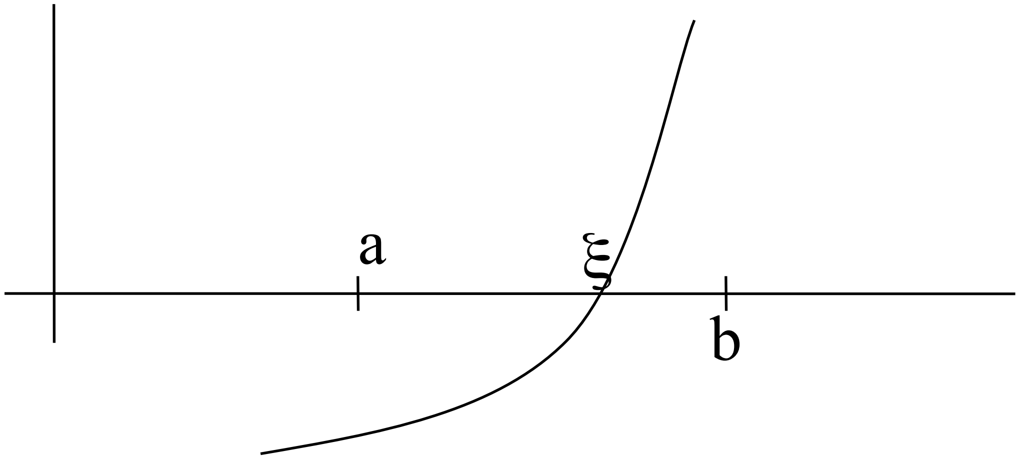
diremos que:

la **razón de convergencia** es, por lo menos, de orden α.

Cuanto mayor sea α y menor sea K, mayor será la velocidad de convergencia

Métodos Cerrados-1

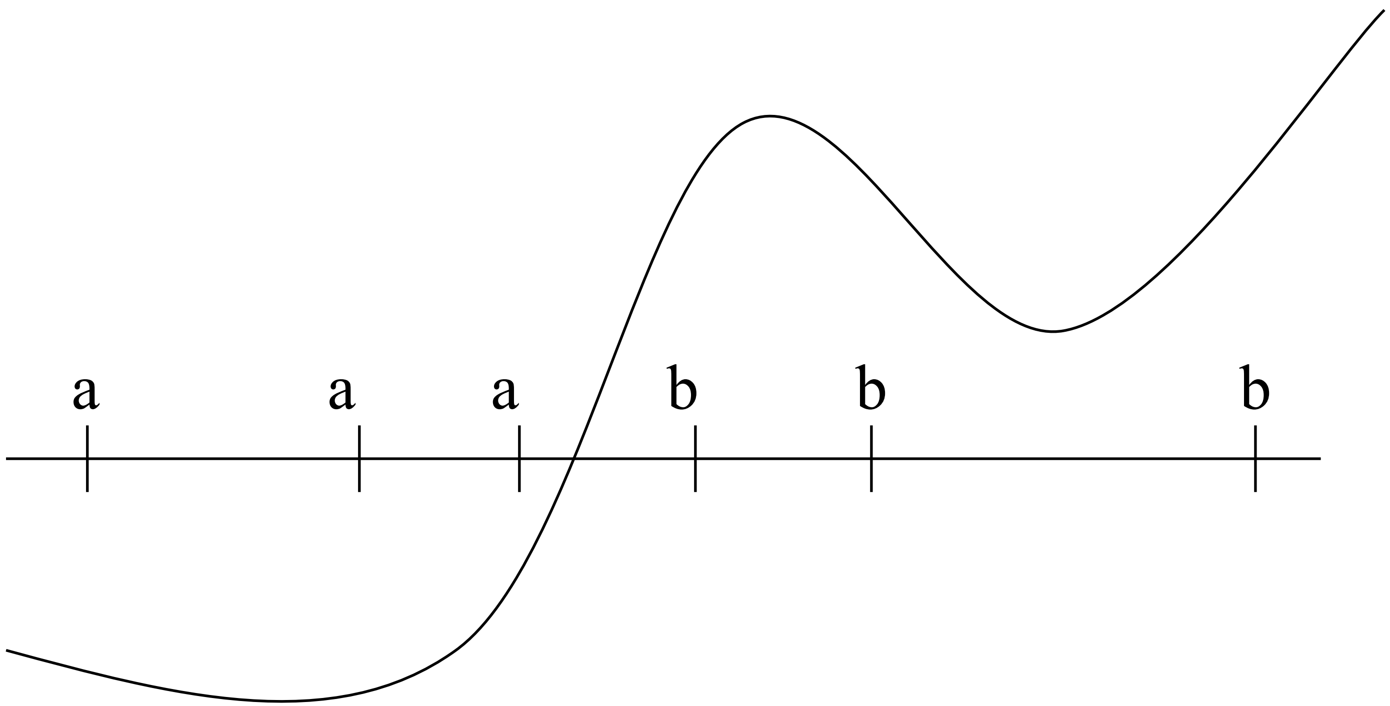
Basados en el teorema de Bolzano:



Si $f(a) \cdot f(b) < 0$, existe al menos un ξ entre a y b tal que
$f(\xi) = 0$

Métodos Cerrados-2

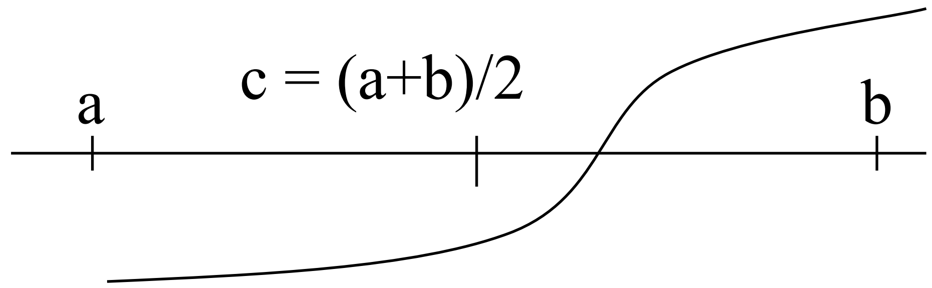
Reducción del intervalo que contiene la raíz, asegurándonos de que sigue conteniéndola en cada paso



DSIC
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

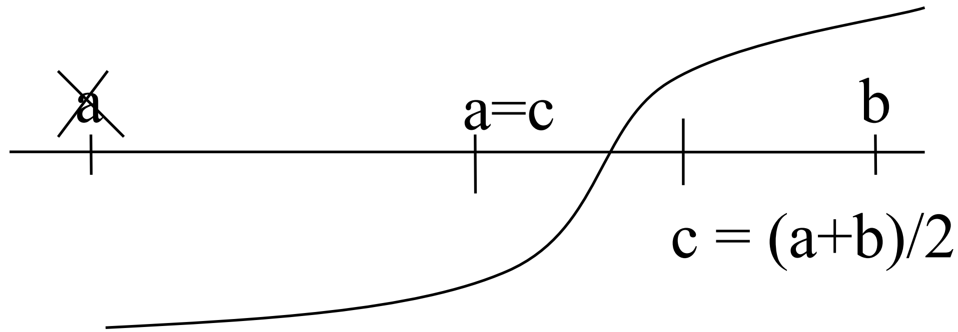
Método de Bisección

1) Cálculo del punto medio del intervalo: $c = (a+b)/2$



2) Se prescinde del extremo del intervalo tal que

$f(\text{extremo}) * f(c) > 0$



DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

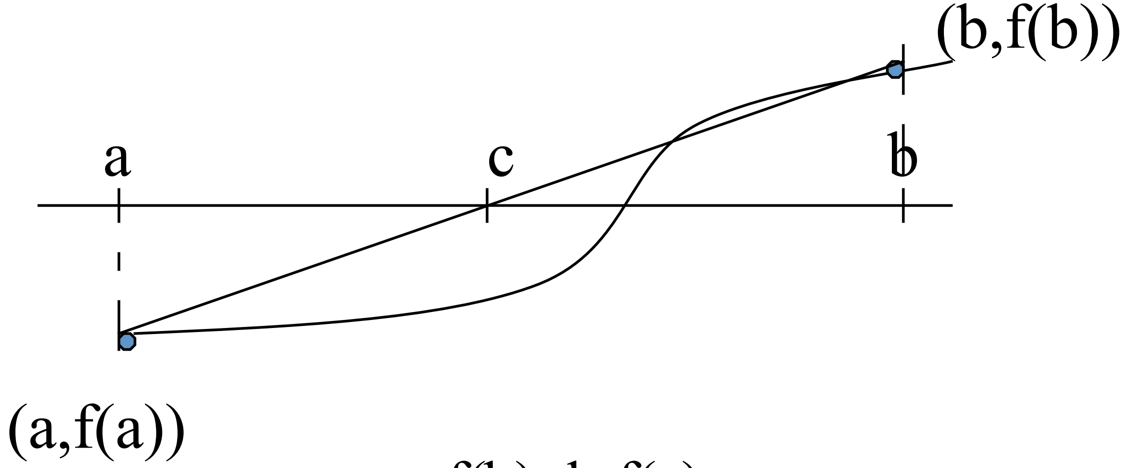
Bisección: Algoritmo

Algoritmo de bisección.

1) Elegir a,b tales que $f(a).f(b) < 0$ y ε
2) Hacer $c = (a+b)/2$
3) Si $|b-c| < \varepsilon \Rightarrow$ c=solución
   si no
   si $f(b).f(c) < 0$
   hacer $a = c$
   si no
   hacer $b = c$
4) Ir a 2.

Regula Falsi

Mejora; en lugar de tomar el punto medio del intervalo, tomamos el punto de corte de la recta que pasa por los puntos (a,f(a)) y (b,f(b)), con el eje de las x.



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

Regula Falsi: Algoritmo

1) Elegir a y b tales que $f(a)\cdot f(b) < 0$ y ε.

2) Hacer $c = \dfrac{a\cdot f(b) - b\cdot f(a)}{f(b) - f(a)}$

3) Si $|b-c| < \varepsilon$ ó $|a-c| < \varepsilon$ entonces

c = solución

si no

si $f(b).f(c) < 0$

hacer a=c

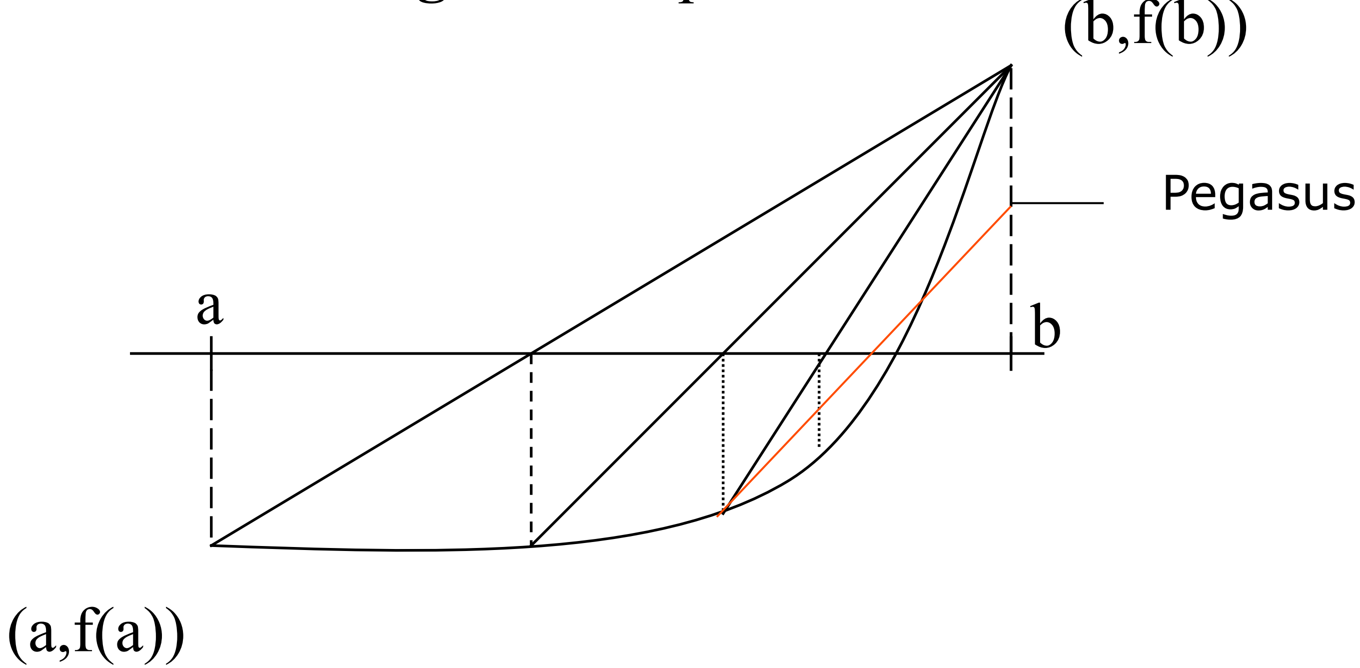
si no

hacer b=c

4) Ir a 2.

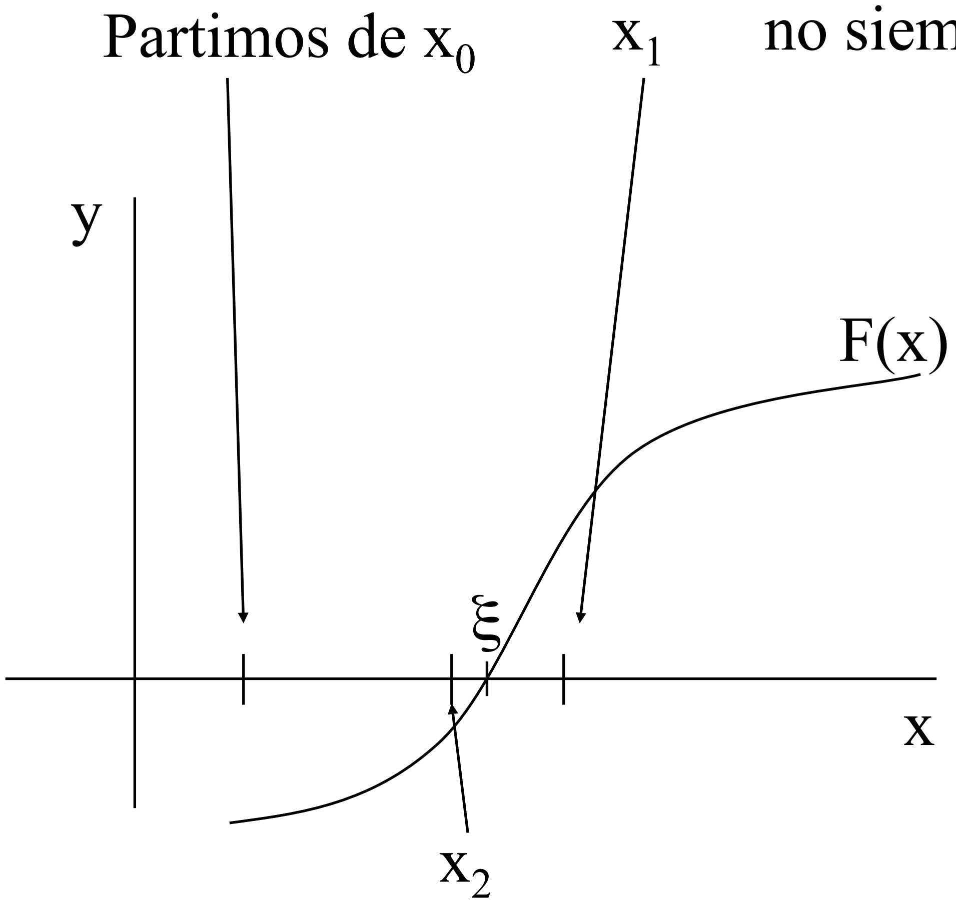
Regula Falsi Modificada

Si la función es cóncava o convexa en todo el intervalo, el rendimiento de **regula falsi** puede disminuir.

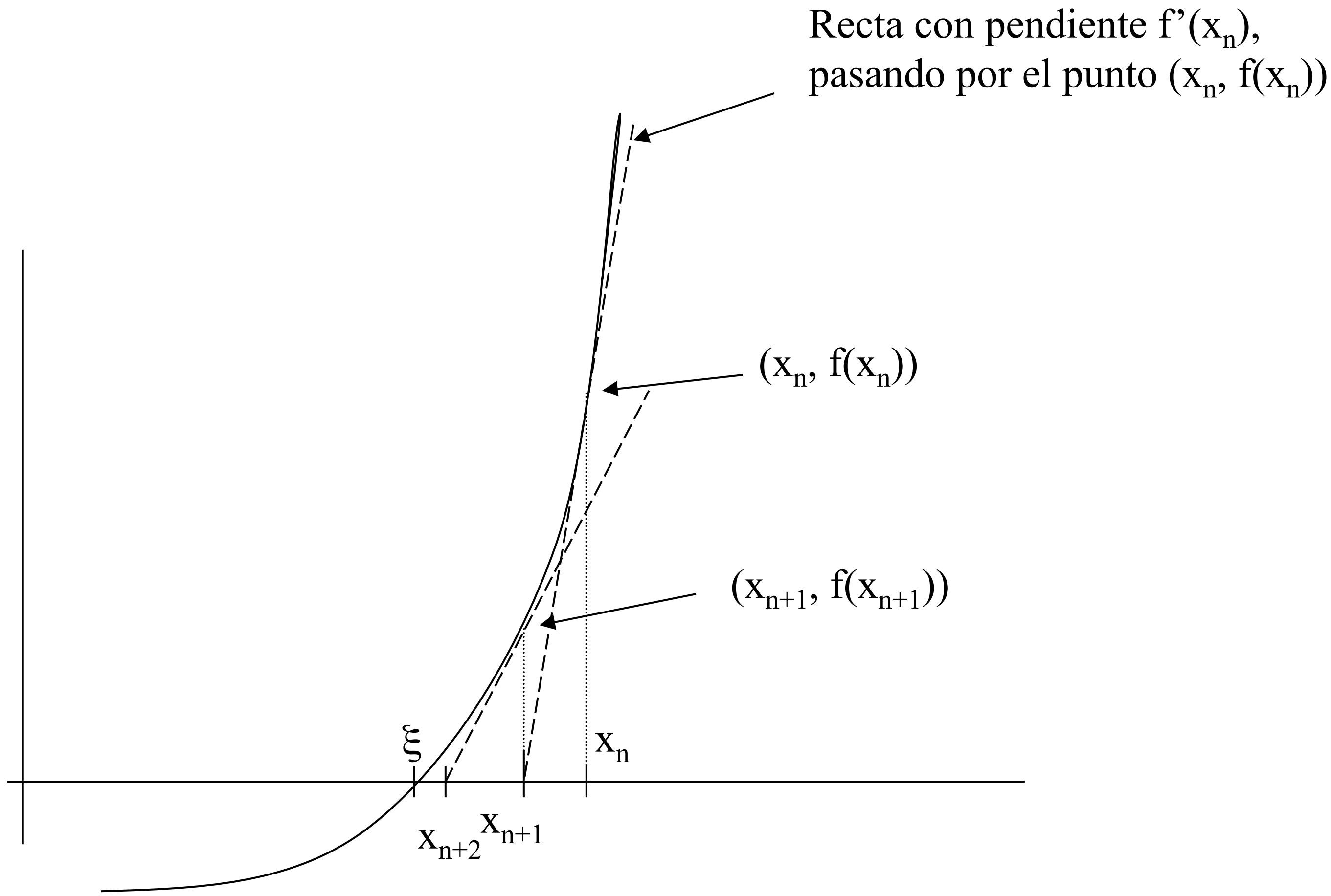


Métodos Abiertos

generamos una sucesión de puntos no basada en el teorema de Bolzano: no siempre convergen



Método de Newton: Interpretación Gráfica



DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Método de Newton -2

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

1) Elegir a, error e I (número máximo de iteraciones).
2) Hacer K = 1 (contador de iteraciones).
3) Hacer b = a - f(a)/f'(a)
4) Mientras |a - b| > error y I>K hacer:
    a = b
    b = a - f(a)/f'(a)
    K = K+1
5) Si K ≥ I entonces no converge
    si no entonces b = solución.

Convergencia **cuadrática**
Convergencia **no garantizada**

Método de Newton -3

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

Alternativa: Mantener la misma derivada durante varias iteraciones

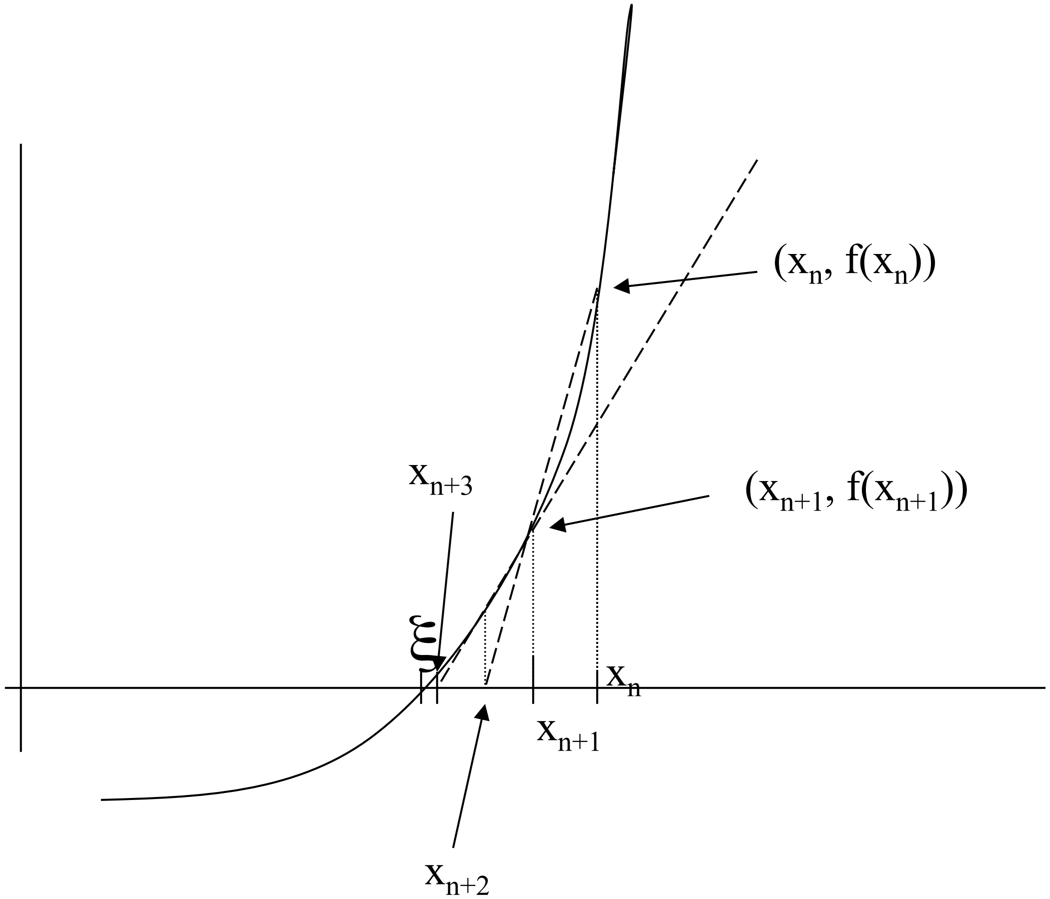
1) Elegir a, error, iter, e I (número máximo de iteraciones).
2) Hacer K = 1 (contador de iteraciones).
3) Hacer b = a - f(a)/f'(a)
4) Mientras |a - b| > error y I>K hacer:
    D=f'(a)
    Para i=1,2,…,iter
        a = b
        b = a - f(a)/D
    Finpara
    K = K+1
5) Si K ≥ I entonces no converge
    si no entonces b = solución.

Convergencia **superlineal**

Convergencia **no garantizada**

Método de la Secante

Se suele aplicar cuando no está disponible la Derivada para aplicar Newton. Se aplica la misma fórmula que en Regula Falsi.



Método de la Secante: Algoritmo

1) Elegir a,b,error e I (número máximo de iteraciones).

2) K = 1 (contador de iteraciones).

3) Hacer $c = \dfrac{a\,f(b) - b\,f(a)}{f(b) - f(a)}$

4) Mientras $|c - b| > \text{error}$ y $I > k$ repetir

$a = b$

$b = c$

$c = \dfrac{a\,f(b) - b\,f(a)}{f(b) - f(a)}$

$K = K + 1$

5) Si $K \geq I$ entonces no converge

si no c = solución.

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Método de la secante

Justificación: $f'(x_n) = \lim_{a \to x_n} \frac{f(x_n) - f(a)}{x_n - a} \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$

Problema:

Como no se requiere que f(b) y f(a) tengan signo distinto, la expresión $c = \frac{a\,f(b) - b\,f(a)}{f(b) - f(a)}$ es propensa a sufrir problemas de error de redondeo

Iteración de Laguerre

Solución de ecuaciones polinómicas con raíces reales:

$$f(x) = 0$$

siendo $f(x) \equiv P_n(x)$ y $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$

las raíces del polinomio.

A partir de una aproximación a la solución, x, obtenemos una nueva aproximación, X, mejor que la anterior.

Paso 1:

Construir una función G(X, x, u) positiva en las raíces y negativa en la aproximación, x, y calcular los puntos de corte, X_1 y X_2 con el eje de abscisas.

Paso 2:

Maximizar/minimizar X_1 y X_2 en función del parámetro u

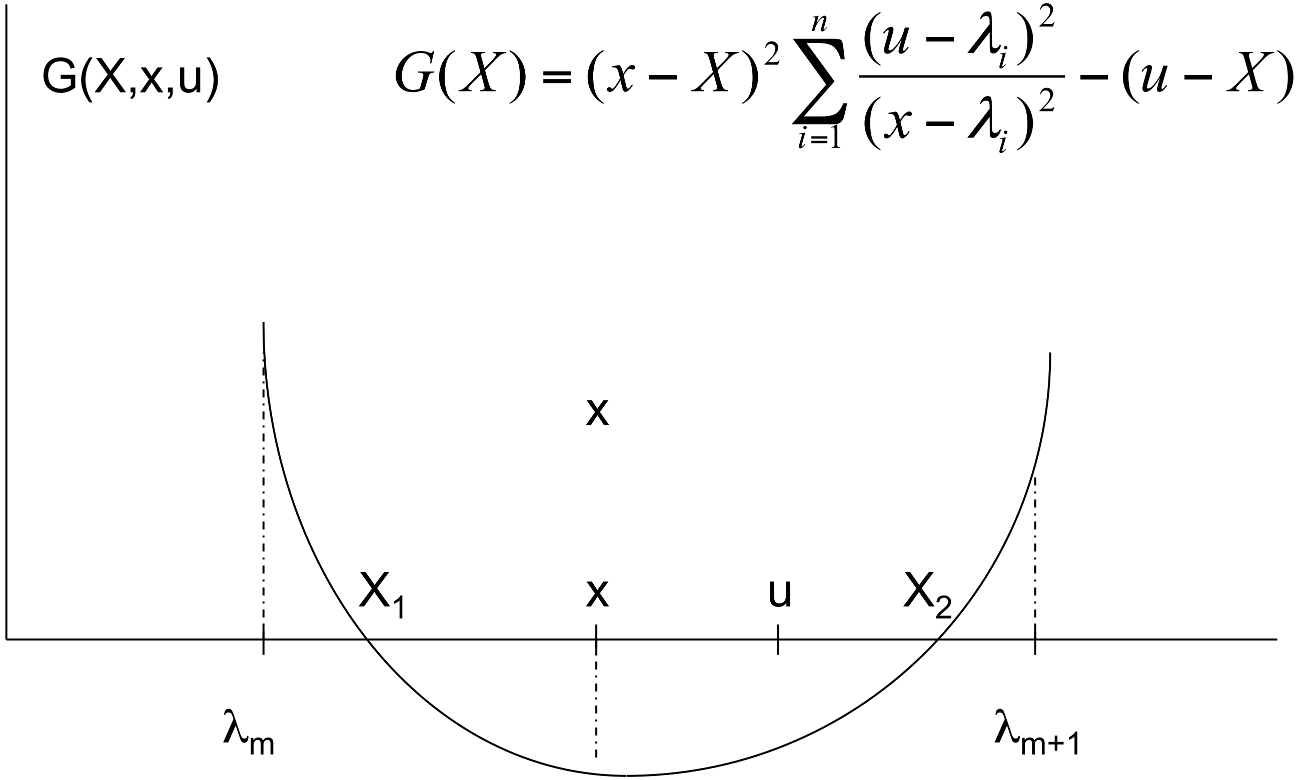
Paso 3:

Escribir la solución óptima encontrada para X_1 y X_2 en función de x, f(x), f' (x) y f''(x).

DEPARTAMENTO DE SISTEMAS
INFORMÁTICOS Y COMPUTACIÓN

Paso 1

$$G(X) = (x - X)^2 \sum_{i=1}^{n} \frac{(u-\lambda_i)^2}{(x-\lambda_i)^2} - (u - X)^2$$



G(X, x, u) puede expresarse como una función cuadrática en la variable u

$$G(X) = Au^2 + 2Bu + C$$

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Si imponemos

$$G(X) = 0 \Rightarrow Au^2 + 2Bu + C = 0$$

con

$$A(X_1) = (x - X_1)^2 \sum_{i=1}^{n} \frac{1}{(x-\lambda_i)^2} - 1$$

$$B(X_1) = (x - X_1)^2 \sum_{i=1}^{n} \frac{-\lambda_i}{(x-\lambda_i)^2} + X_1$$

$$C(X_1) = (x - X_1)^2 \sum_{i=1}^{n} \frac{\lambda_i^{\,2}}{(x-\lambda_i)^2} - X_1^{\,2}$$

y expresiones análogas para X_2 .

De aquí se puede despejar X_1 y X_2 en función de u, o bien considerar que se han definido de forma implícita en función de u.

Paso 2

Para optimizar X_1 y X_2 imponemos sobre

$$Au^2 + 2Bu + C = 0$$

las condiciones

$$\frac{dX_1}{du} = 0 \quad y \quad \frac{dX_2}{du} = 0$$

Las derivadas se calculan de forma implícita.
Se obtiene como u_{opt} el valor:

$$u_{opt} = -\frac{B}{A}$$

y para X_1 y X_2 valores que se pueden obtener de

$$AC - B^2 = 0$$

Es decir:

$$X_1 = X_1(x, \lambda_1, \lambda_2, \ldots, \lambda_n)$$

$$X_2 = X_2(x, \lambda_1, \lambda_2, \ldots, \lambda_n)$$

Paso 3

Por último basta poner $\lambda_1, \lambda_2, \ldots, \lambda_n$

en función de f(x), f´(x) y f´´(x) .

Puesto que $f(x) = a_n \prod_{i=1}^{n} (x - \lambda_i)$ **derivando y operando se tiene**

$$\sum_{i=1}^{n} \frac{1}{(x-\lambda_i)^2} = \left[\frac{f'(x)}{f(x)}\right]^2 - \frac{f''(x)}{f(x)}$$

$$\sum_{i=1}^{n} \frac{\lambda_i^{\,2}}{(x-\lambda_i)^2} = x^2\left(\left[\frac{f'(x)}{f(x)}\right]^2 - \frac{f''(x)}{f(x)}\right) - 2x\left(\frac{f'(x)}{f(x)}\right) + n$$

$$\sum_{i=1}^{n} \frac{-\lambda_i}{(x-\lambda_i)^2} = -x\left(\left[\frac{f'(x)}{f(x)}\right]^2 - \frac{f''(x)}{f(x)}\right) + \left(\frac{f'(x)}{f(x)}\right)$$

Por último, sustituyendo en

$$AC - B^2 = 0$$

se tiene

$$X = x - \frac{n}{\left(\frac{f'(x)}{f(x)}\right) \mp \sqrt{(n-1)^2\left(\frac{f'(x)}{f(x)}\right)^2 - n(n-1)\left(\frac{f''(x)}{f(x)}\right)}}$$

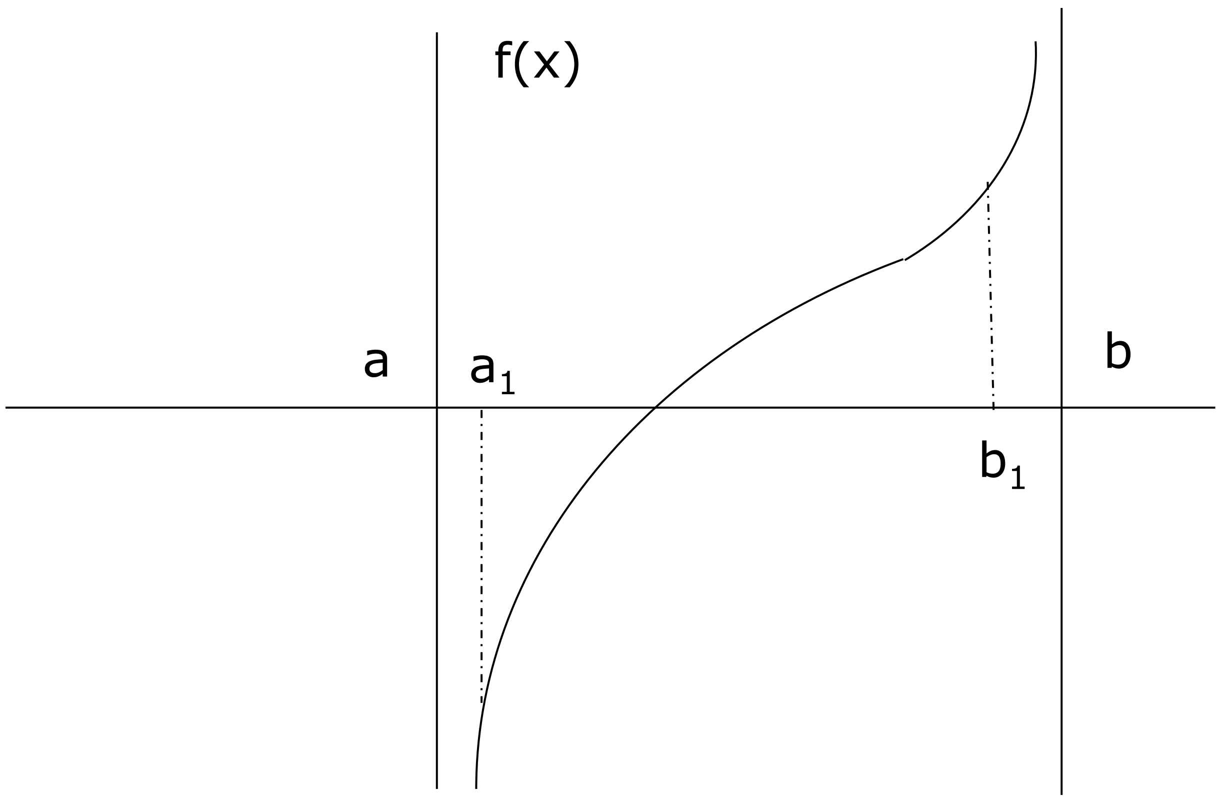
Esta es la iteración de Laguerre que converge de forma cúbica a la solución de la ecuación polinómica si existe solución, el punto inicial está lo suficientemente próximo a la solución, y existen las derivadas.

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Interpolación racional

Idea: Aproximar f(x) por una función racional lo más parecida posible

$$f(x) \cong \varphi(x) = \frac{K_1 x + K_2}{(x-a)(x-b)}$$



DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Conociendo dos puntos más se puede calcular K_1 y K_2

$$\varphi(a_1) = \frac{K_1 a_1 + K_2}{(a_1 - a)(a_1 - b)} \qquad \varphi(b_1) = \frac{K_1 b_1 + K_2}{(b_1 - a)(b_1 - b)}$$

O bien
$$\begin{cases} K_1 a_1 + K_2 = \alpha_1 \\ K_1 b_1 + K_2 = \alpha_2 \end{cases} \quad \text{con} \quad \begin{cases} \alpha_1 = f(a_1)(a_1 - a)(a_1 - b) \\ \alpha_2 = f(b_1)(b_1 - a)(b_1 - b) \end{cases}$$

Por tanto
$$\begin{cases} K_1 = \dfrac{\alpha_1 - \alpha_2}{a_1 - b_1} \\ K_2 = \alpha_1 - K_1 a_1 \end{cases}$$

Si se aproxima $f(x) \cong \varphi(x)$ se puede tomar $\varphi(c) = 0$

para averiguar el nuevo punto de corte

Algoritmo

Elegir a_1 y b_1 tales que $f(a_1)f(b_1) < 0$

Calcular K_1 y K_2

Repetir

$$c = {-K_2}/{K_1}$$

Si $f(c)f(b_1) > 0$

entonces

$$b_1 = c$$

en otro caso

$$a_1 = c$$

Recalcular K_1 y K_2

hasta que $|b_1 - a_1| < \text{cota}$

Ventajas:

Convergencia muy rápida porque la aproximación al a función es muy buena

Inconveniente:

Sólo es útil con este tipo de funciones

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Ejercicios propuestos

- Escribir un programa en Matlab que calcule las raices de una función polinómica de grado m, utilizando:
- 1. Método de bisección
- 2. Método “Regula falsi”
- 3. Método de Newton
- 4. Método “chord” (Newton con derivada constante)