Raíces de ecuaciones no lineales 2022

Prof.: Dr. Juan Ignacio Manassaldi

J.T.P.: Ing. Amalia Rueda

Introducción

- La determinación de las raíces de una ecuación algebraica racional entera (polinomio) es uno de los problemas más antiguos de las matemáticas que se presentan con frecuencia en la resolución de problemas reales.
- Existe una gran variedad de métodos de resolución que prueban la larga historia en el análisis de este problema y de su importancia hasta la actualidad.
- Estos métodos se diferencian por la necesidad de:
  - Obtener todas las raíces de una ecuación o únicamente algunas de ellas.
  - Determinar todas las raíces reales o complejas, simples o múltiples.
  - De disponer de una 1era. aproximación para c/u de ellas.
- Disponiéndose en la actualidad de computadoras digitales, resulta conveniente utilizar métodos más apropiados para obtenerlas.

Introducción

- Dado el caso de funciones racionales enteras (polinomios) igualadas a cero de la forma:

$$P_n\left(x\right) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_1 x + a_0 = 0$$

con $a_n \neq 0$ y n entero > 0 y $a_i \in R$ con $i = 0, 1, 2, 3, \ldots, n$ constantes, existen métodos para resolverlas directamente en términos de sus coeficientes, siempre que el grado n de la ecuación sea a lo sumo 4.

- Las llamadas “resolventes” solo existen hasta polinomios de grado 4, siendo las de grado 3 y 4 de elevada complejidad computacional.

Conclusiones

- Para las ecuaciones **algebraicas no lineales** (en particular para los polinomios de grado $n > 4$), para las **ecuaciones trascendentes** y aún para las ecuaciones **polinómicas de tercer y cuarto grado** (por lo complicado de las resolventes) conviene establecer métodos generales que nos permitan resolverlas aunque más no sea de una forma aproximada.
- Paradójicamente, la solución aproximada en la mayoría de los casos es mejor que la que suministran los métodos directos (sólo en el caso de los polinomios).
- Los métodos aproximados permiten mejorar la solución por simple repetición del mismo método hasta lograr el grado de aproximación requerido.

Raíces de ENL

Solución numérica de ecuaciones algebraicas y trascendentes de una variable

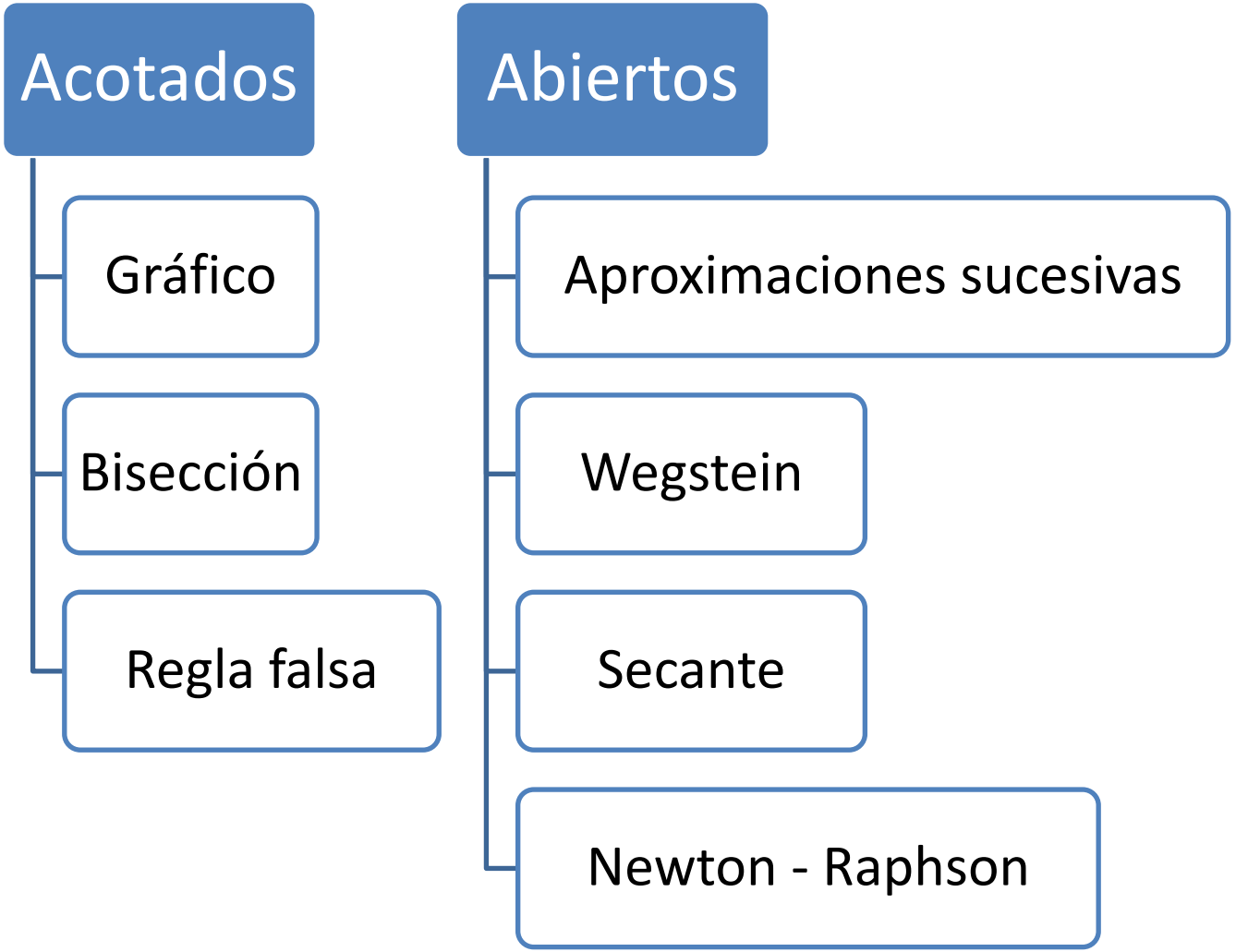
- Sea una función de una variable $f(x)$.
- Se busca encontrar un valor x^* para el que se cumpla $f(x^*) = 0$.
- Si existe este valor se lo denomina raíz de la ecuación.
- Pasos básicos para encontrarlo:
  - Determinación de un valor aproximado de la raíz (valor de arranque de método).
  - Mejoramiento de la solución hasta un grado de precisión establecido.

tolerancia

Interpretación del problema

Método iterativo de calculo

Métodos de resolución de ENL



Métodos iterativos

- **Solución Iterativa**: Significa comenzar con una solución inicial (aproximada) y generar una secuencia de números (sucesión):

$$\{x_n\} = x_1, x_2, ..., x_n$$

tal que si existe

$$\lim_{n \to \infty} \{x_n\} = x^* \quad y \quad f(x^*) = 0$$

entonces x^* es una raíz de la ecuación $f(x^*) = 0$.

- Error Absoluto Exacto en la Iteración n:

$$E_{exacto}^{(n)} = |x_n - x^*|$$

Métodos iterativos

- El error exacto en la iteración ***n*** no se conoce:

$$E_{exacto}^{(n)} = \left| x_n - x^* \right|$$

por lo tanto no puede utilizarse como criterio de terminación de método.

- Tolerancia del Error:

$$\left| f\left(x_n\right) \right| < \varepsilon$$

$$E_{aprox.}^{(n)} = \left| x_n - x_{n-1} \right| < \varepsilon$$

Error absoluto

$$e_{rel.}^{(n)} = \left| \frac{x_n - x_{n-1}}{x_n} \right| < \varepsilon$$

Error relativo

$$si : x_n \to x^* \Rightarrow f\left(x_n\right) \to 0$$

Métodos iterativos

Orden de convergencia

- Sea $\{x_n\}$ una secuencia que converge a x^*.
- Si existen las constantes positivas ***p*** y ***K*** tales que:

$$\lim_{n\to\infty} \frac{\left|x_{n+1} - x^*\right|}{\left|x_n - x^*\right|^p} = \lim_{n\to\infty} \frac{E_{exacto}^{(n+1)}}{\left(E_{exacto}^{(n)}\right)^p} = K$$

- Entonces, se dice que la secuencia $\{x_n\}$ converge a x^* con orden ***p*** y un un error asintótico representado por la contante ***K***.
- Una secuencia con mayor orden de convergencia convergerá más rápido que una con bajo orden.
- Por otro lado, además del orden de convergencia, en el proceso de cálculo interviene el costo computacional por iteración que es importante para definir la eficiencia del método.

Métodos iterativos

Método	Orden de Convergencia
Aproximaciones sucesivas	Lineal (p=1)
Newton - Raphson	Cuadrático (p= 2)
Secante	Superlineal (p = 1.618)
Bisección	Lineal (p =1)

Objetivos

Teóricos/Prácticos

- Entender y aplicar los algoritmos de manera manual.
- Interpretar problemas de ingeniería y proponer una aproximación de la solución.
- Programar una “function” de scilab para cada método de manera de poder aplicarlas en los diferentes problemas de la guía.
- Plantear y resolver problemas de ingeniería.

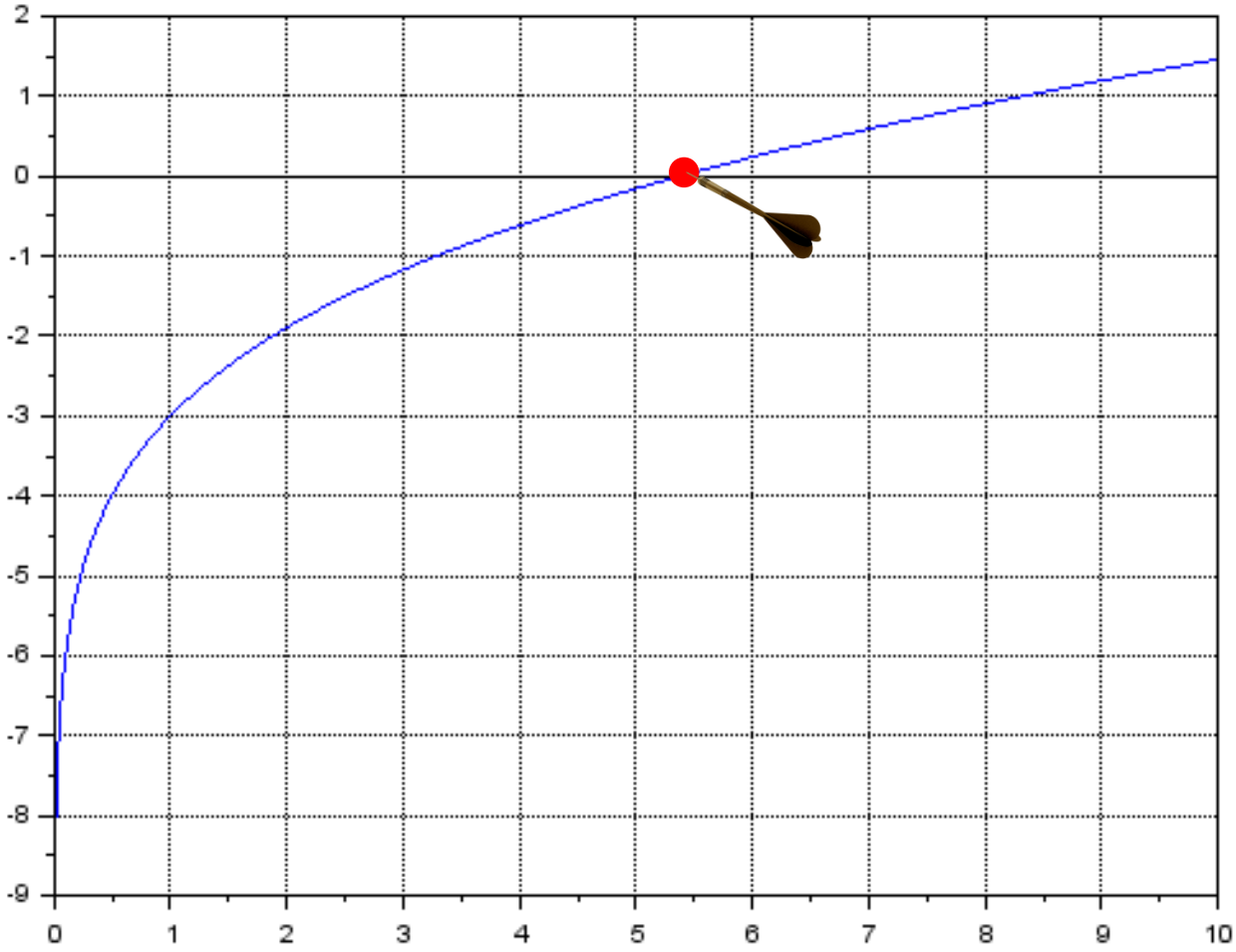
Ejemplo

$$\sqrt{x} + \ln(x) - 4 = 0$$

```
function y=f(x)
y=sqrt(x) + log(x) - 4;
endfunction
```

Método Gráfico

Para utilizar este método graficamos la ecuación y estimamos visualmente donde se encuentra la raíz

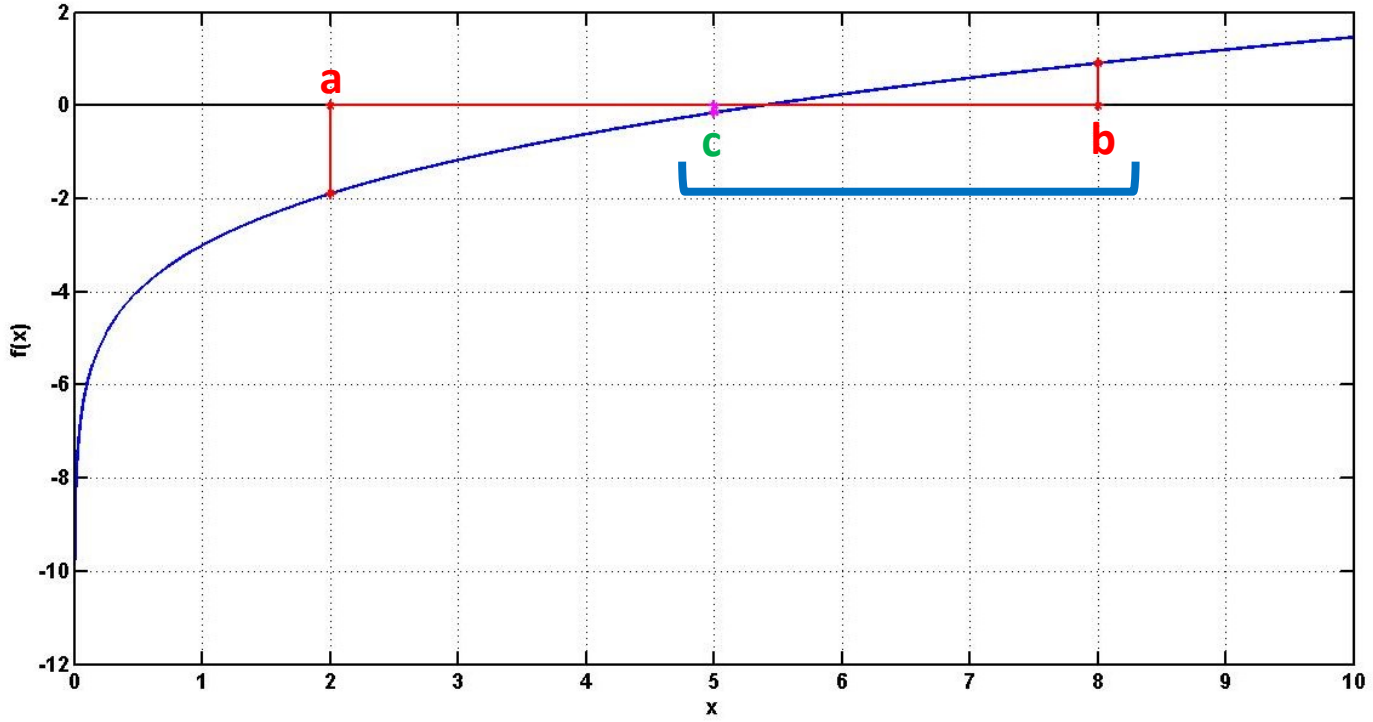



```
x=linspace(0,10,1000);
plot(x,f(x),'b')
plot(x,zeros(1:1000),'k')
xgrid
```

```
--> xclick
 ans  =

   3.   5.3278689   0.0072464
```

Método de la Bisección

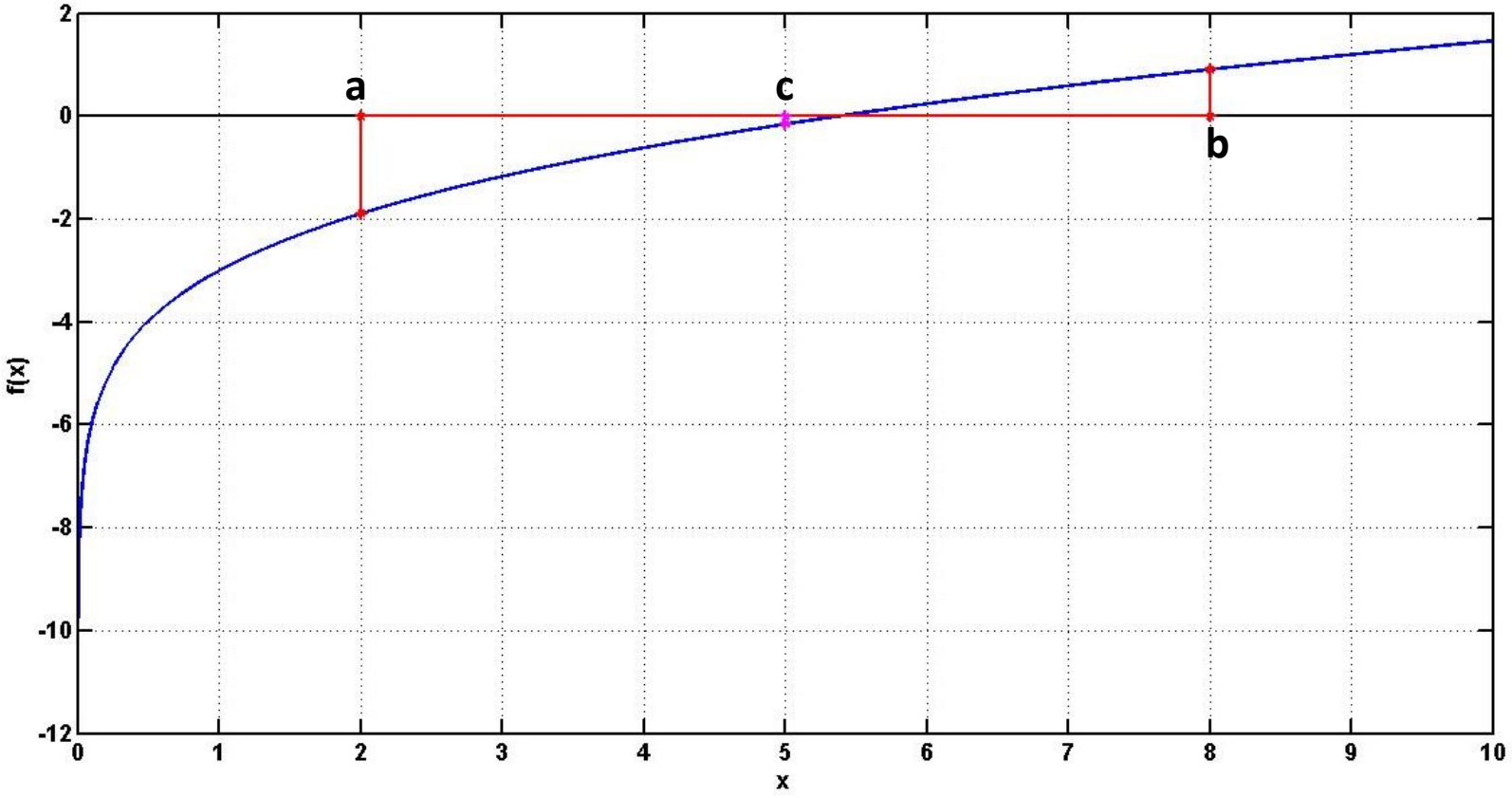
Para utilizar este método necesitamos dos puntos donde la función cambie de signo. El nuevo punto corresponde a la mitad del intervalo y debe seleccionarse de manera de conservar la diferencia de signo.



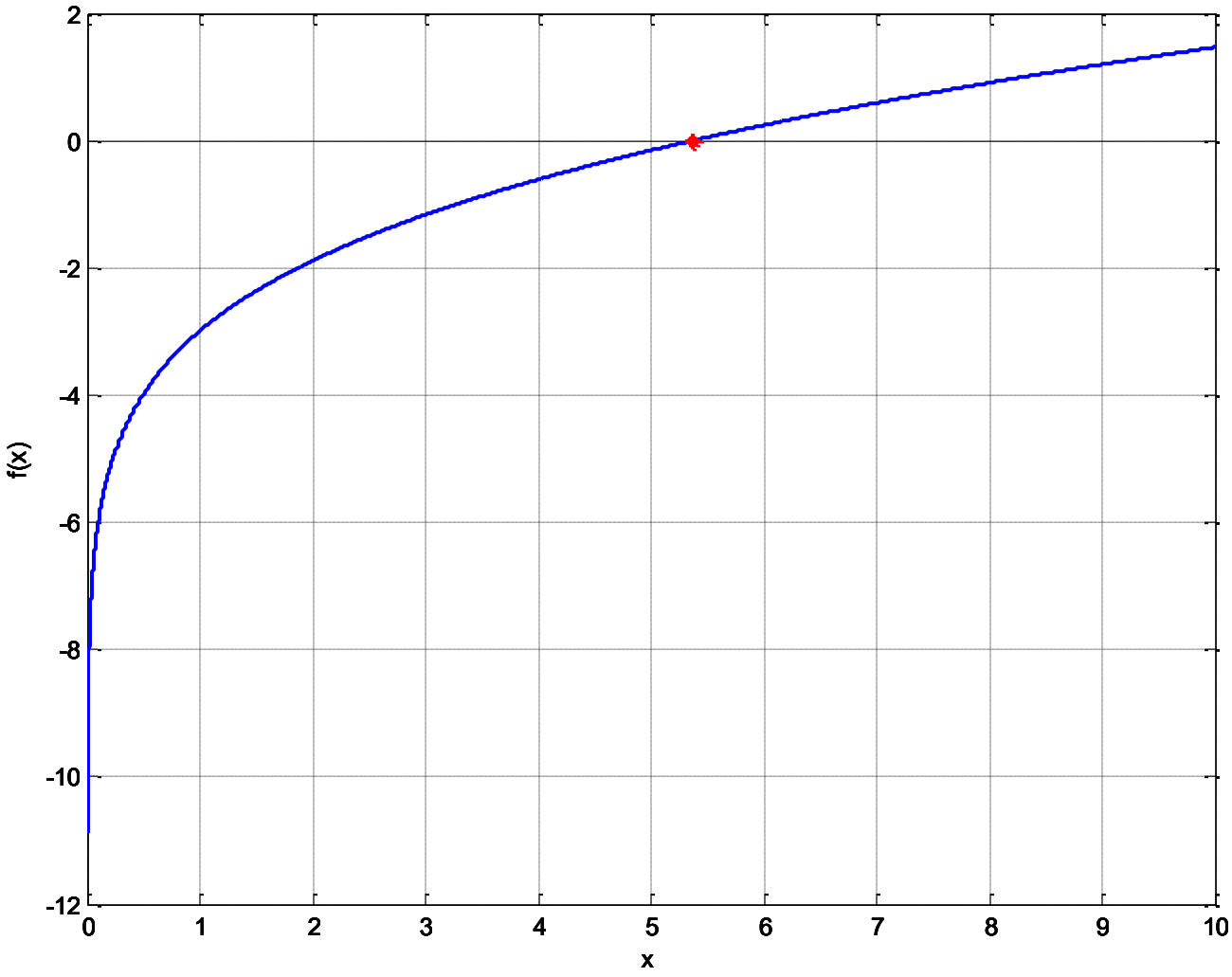
Método de la Bisección

Debemos lograr que el intervalo sea lo mas pequeño posible en torno a la raíz.
¿Cómo lo achicamos?

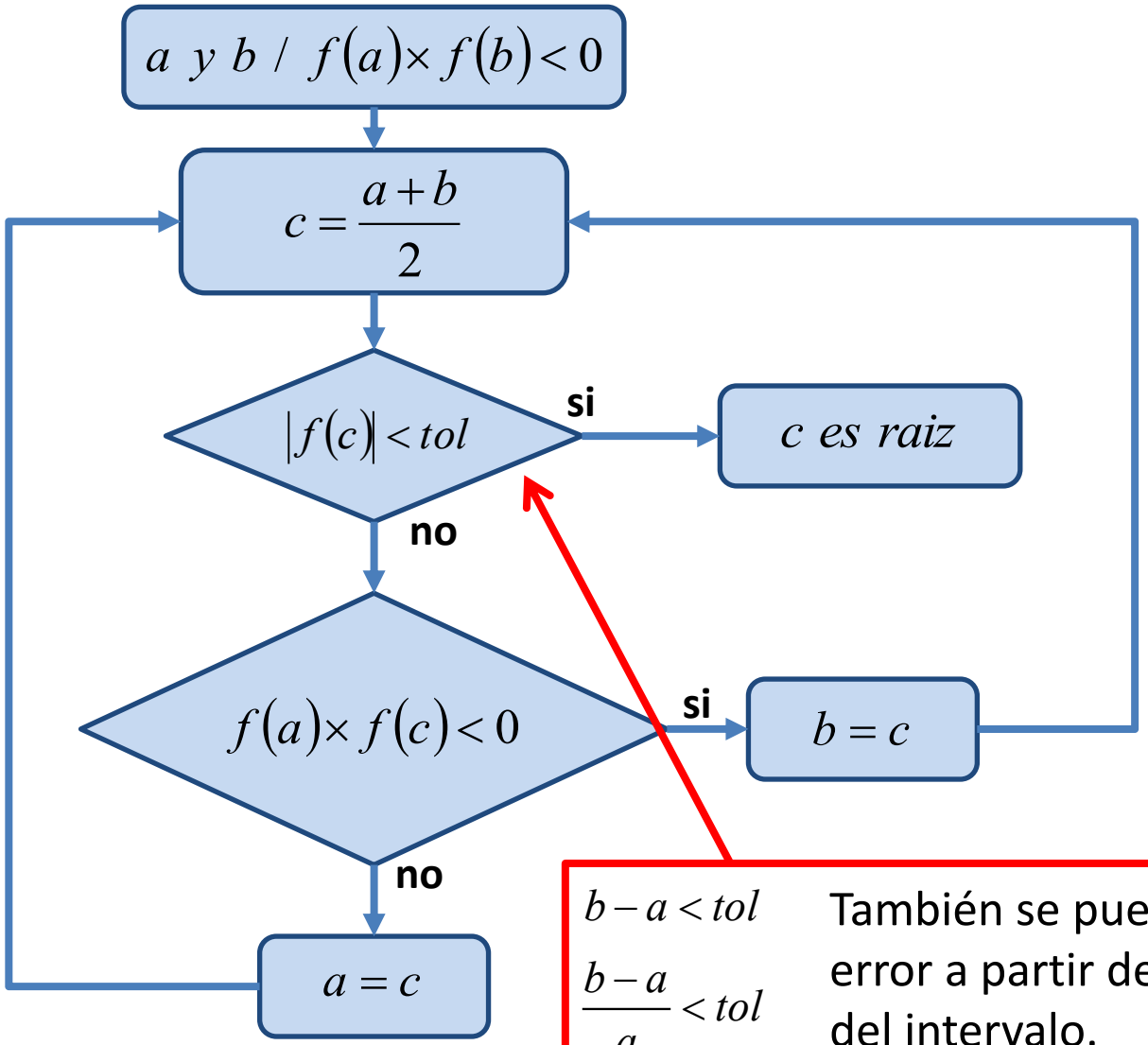
Evaluamos el valor de la función en la mitad del intervalo
¿Cuál es el nuevo intervalo?



Método de la Bisección



Método de la Bisección



$b-a<tol$

$\frac{b-a}{a}<tol$

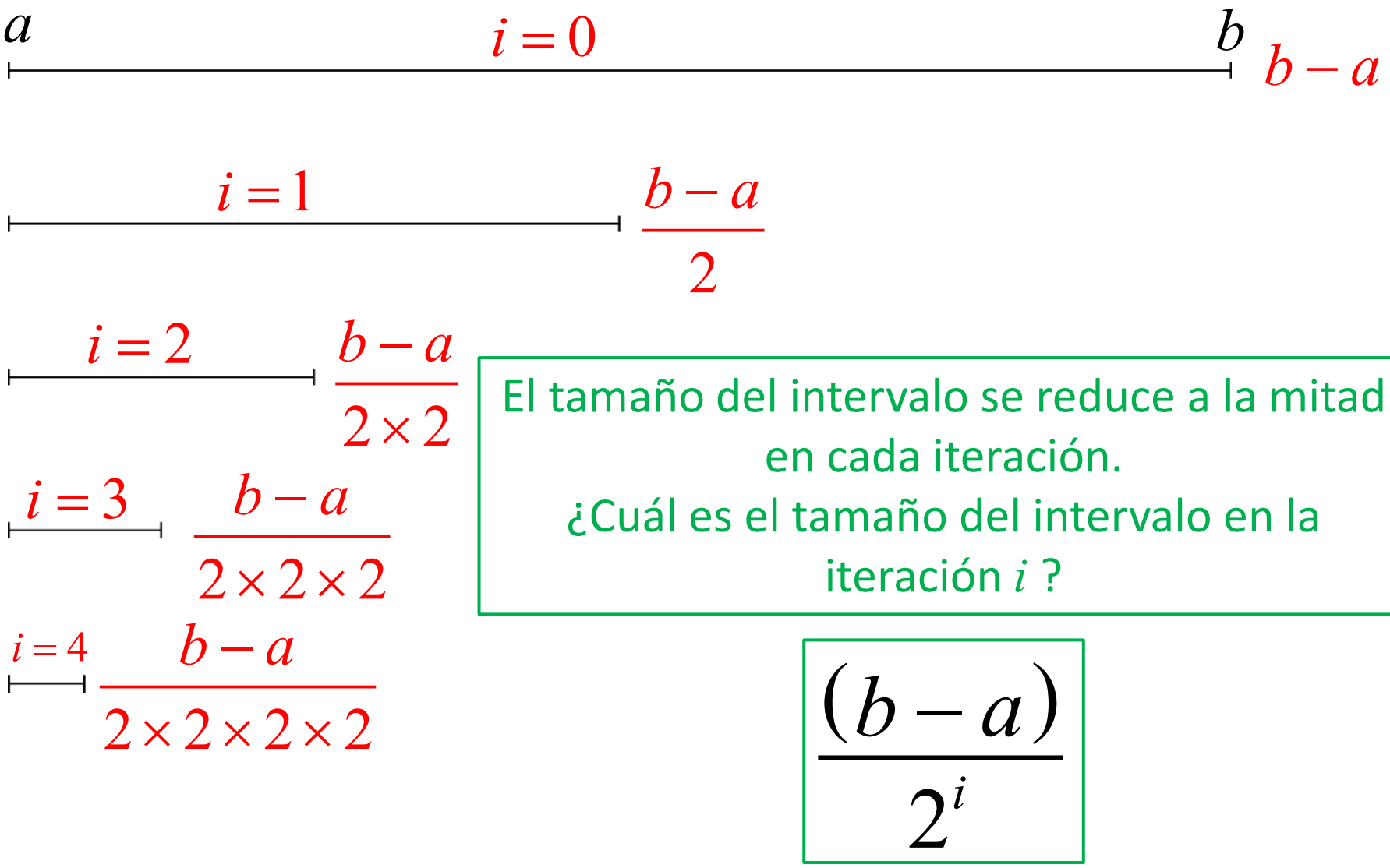
También se puede medir el error a partir del tamaño del intervalo.

Método de la Bisección

a	b	c	f(a)	f(b)	f(c)	error

$$\sqrt{x} + \ln(x) - 4 = 0$$

Método de la Bisección



El tamaño del intervalo se reduce a la mitad en cada iteración.
¿Cuál es el tamaño del intervalo en la iteración i ?

$$\frac{(b-a)}{2^i}$$

Método de la Bisección

La relación del tamaño del intervalo con el número de iteración nos sirve para conocer de antemano cuantas iteraciones debemos realizar.

Es decir, en la iteración N el tamaño del intervalo debe ser igual a la tolerancia deseada.

$$tol = \frac{(b-a)}{2^N}$$

Método de la Bisección

Ejemplo:

Encontrar el numero de iteraciones necesarias para obtener por el método de la bisección la raíz de la ecuación presentada en el intervalo [5 6] adoptando una tolerancia de $tol=10^{-5}$.

$$tol = \frac{(b-a)}{2^N} \Rightarrow 10^{-5} = \frac{(6-5)}{2^N}$$

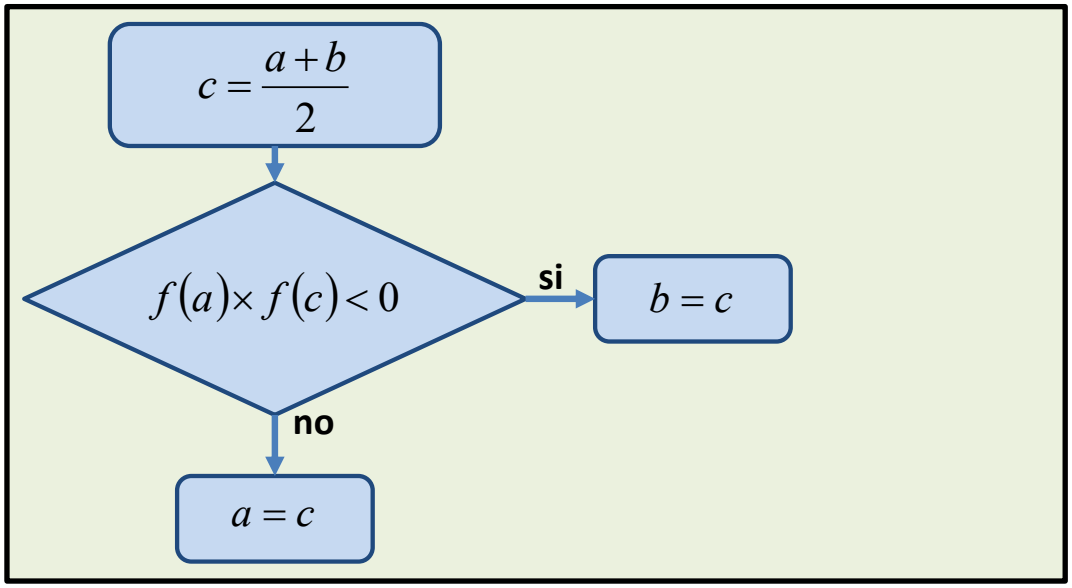
$$-5 = \log_{10}(6-5) - N\log_{10} 2$$

$$N = \frac{\log_{10}(6-5)+5}{\log_{10} 2} \qquad \Rightarrow N = 16.609$$

$$(17 \text{ iteraciones})$$

Método de la Bisección (Scilab)

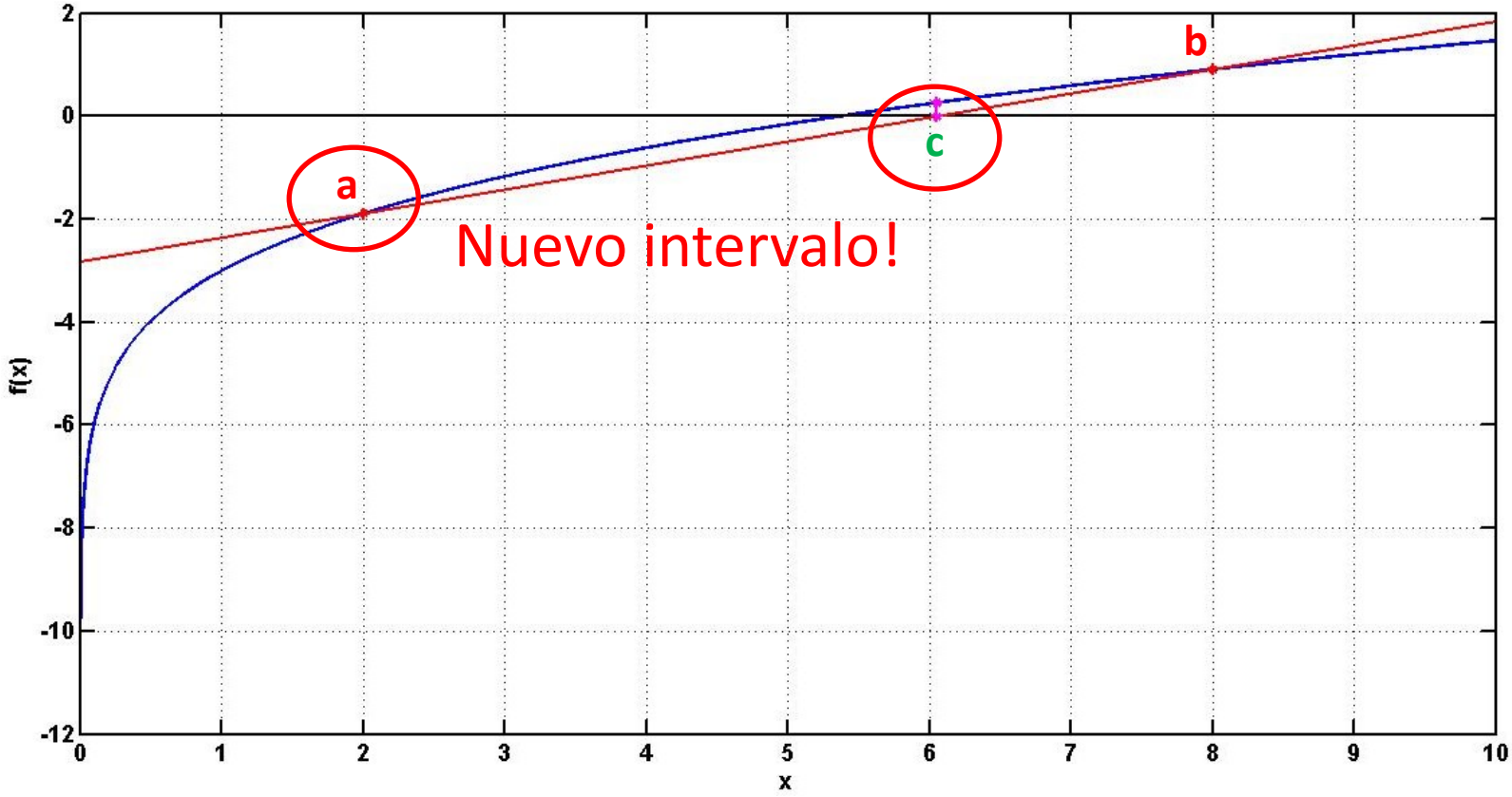
```
function out=biseccion(fun, a, b, tol)
N=ceil((log(b-a) - log(tol))/log(2));
for k=1:N
```



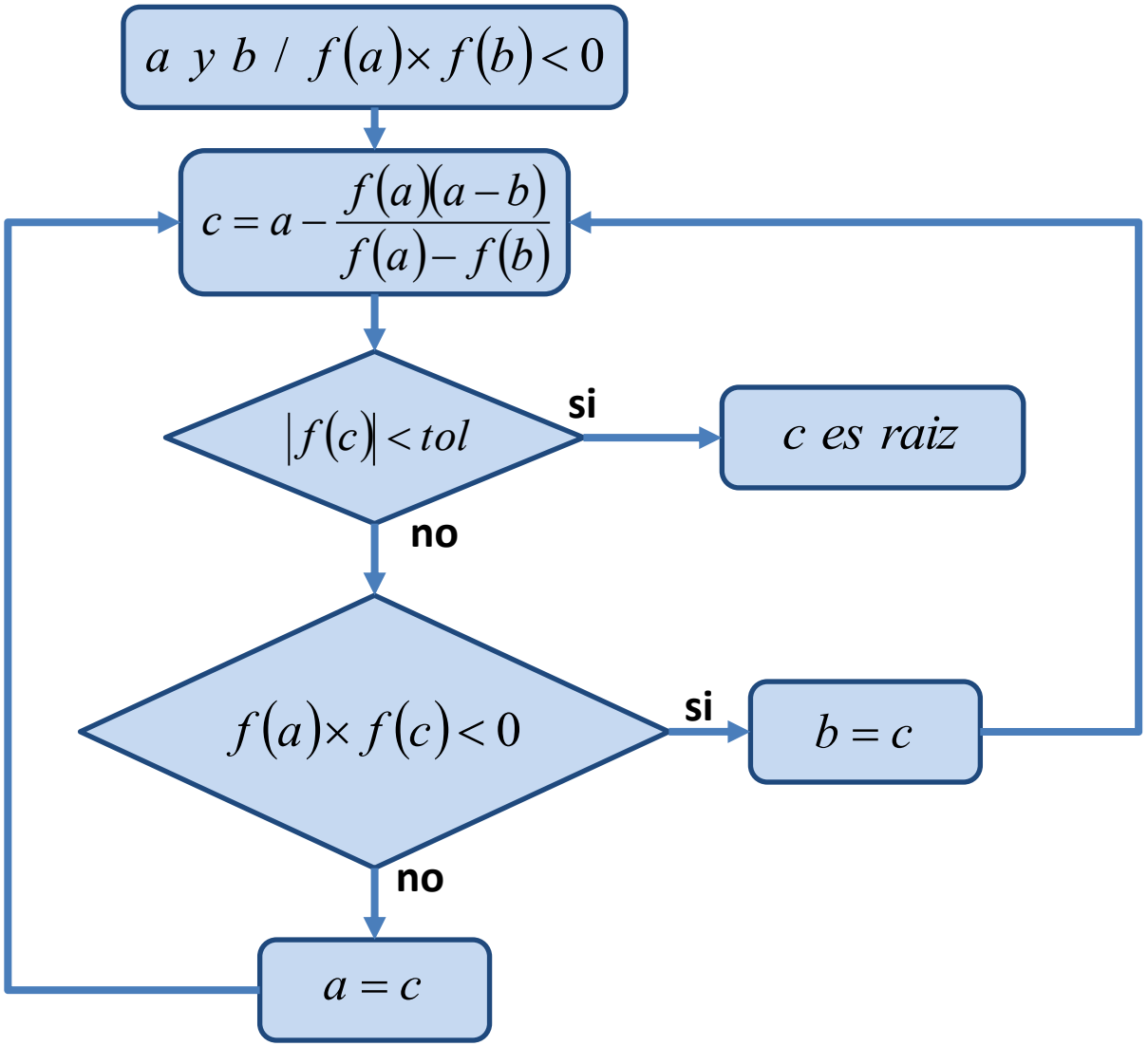

```
end
out=c;
endfunction
```

Método de la Regla Falsa

El proceso es similar al de bisección, pero el nuevo punto no corresponde a la mitad del intervalo sino a la intersección de la recta que los une (secante) con el eje de las abscisas.



Método de la Regla Falsa



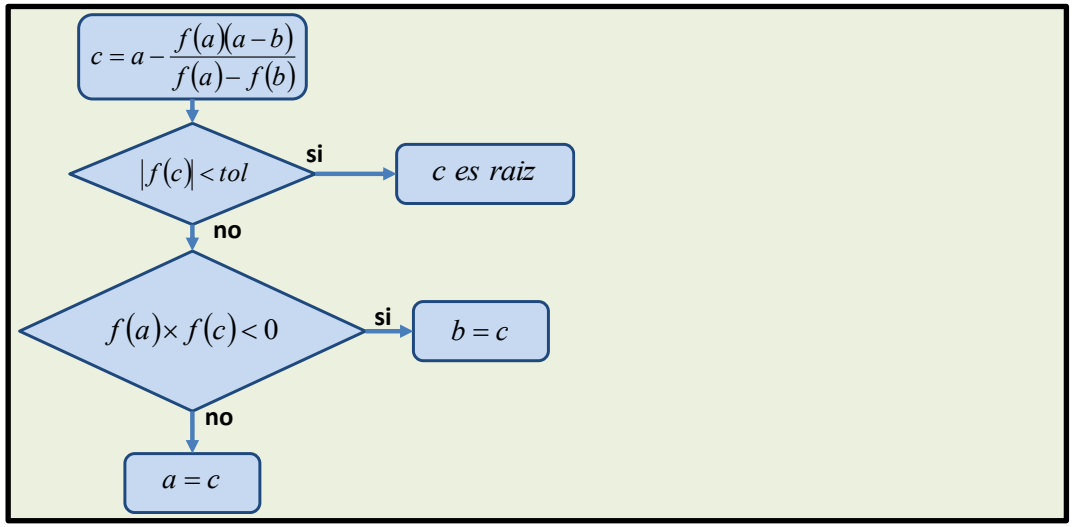
Método de la Regla Falsa

a	b	f(a)	f(b)	c	error

$$f(x) = \ln(x) + e^{2x}$$

Método de la Regla Falsa (Scilab)

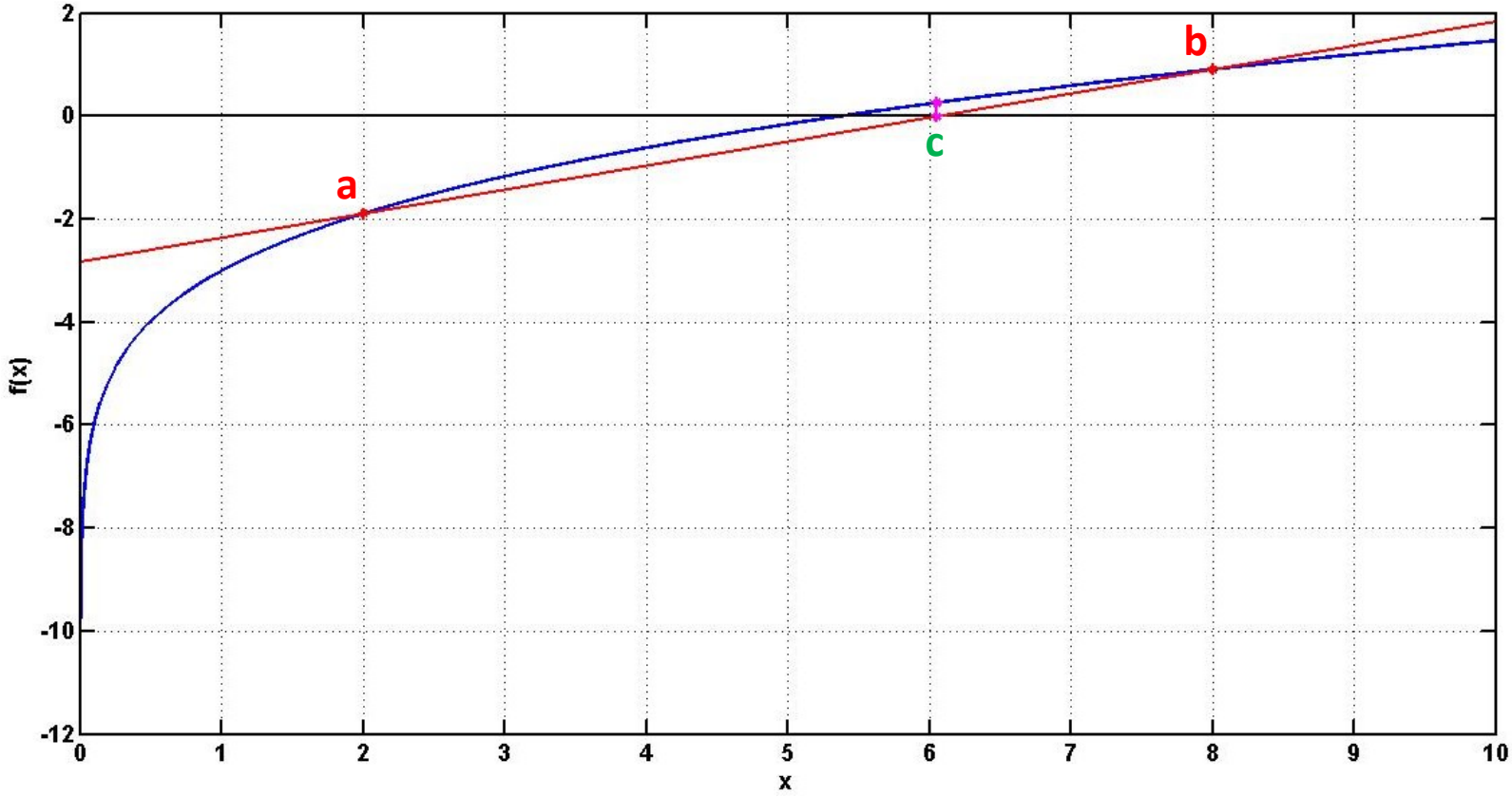
```
function out=reglafalsa(fun, a, b, tol)
for k=1:100
```



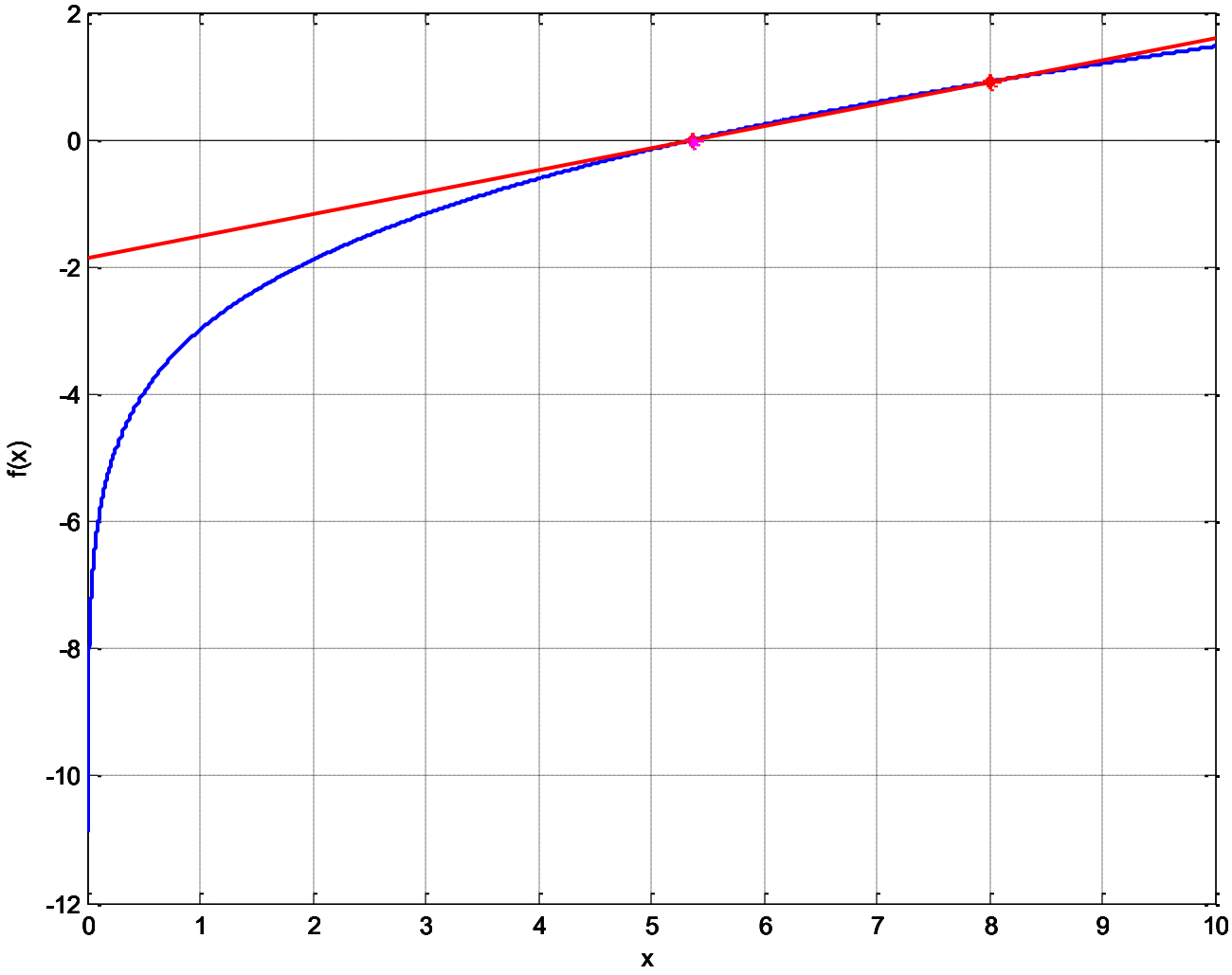

```
end
if k == 100
        out=[];
        disp('aumentar iteraciones');
end
endfunction
```

Método de la Regla Falsa (Punto fijo)

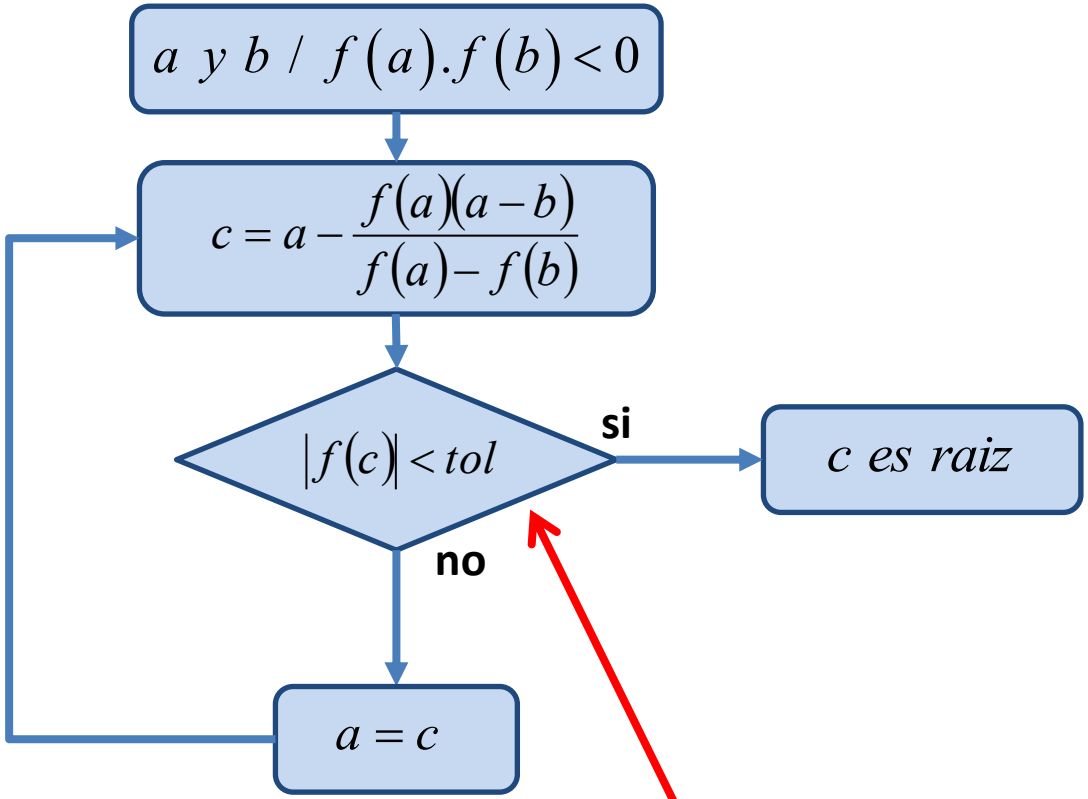
Para utilizar este método necesitamos dos puntos donde la función cambie de signo. El nuevo punto surge de la intersección de la recta que los une con el eje de las abscisas.



Método de la Regla Falsa (Punto fijo)



Método de la Regla Falsa (Punto fijo)



$|c-a| < tol$

$\left|\frac{c-a}{a}\right| < tol$

También se puede medir el error a partir del valor de la variable independiente

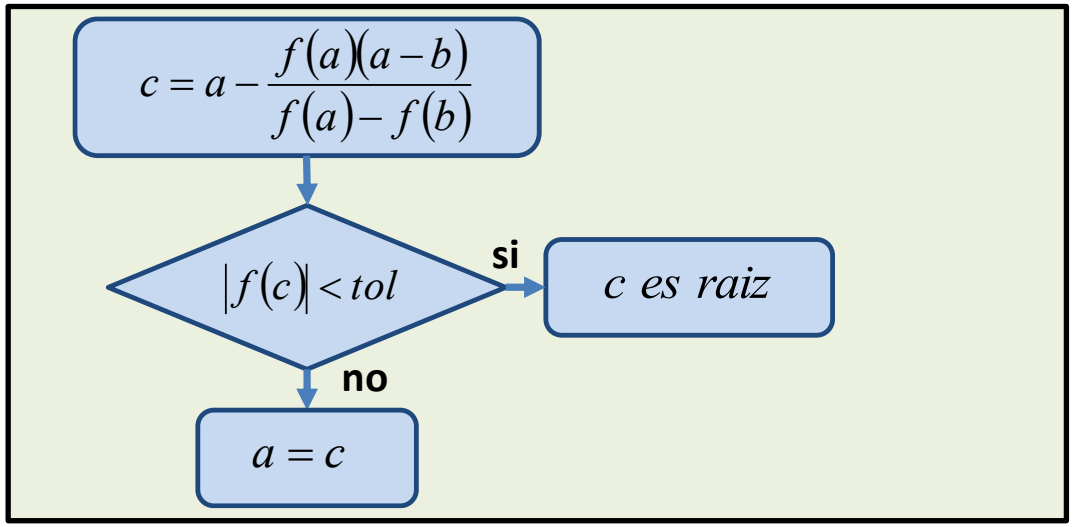
Método de la Regla Falsa (Punto fijo)

a	b	f(a)	f(b)	c	error

$$f(x) = \ln(x) + e^{2x}$$

Método de la Regla Falsa (Punto fijo - Scilab)

```
function out=reglafalsapf(fun, a, b, tol)
for k=1:100
```




```
end
if k == 100
        out=[];
        disp('no converge’);
end
endfunction
```

Métodos abiertos

A los fines de su consideración en la resolución de problemas de ingeniería, podemos agruparlos en dos categorías:

- Método de Aproximaciones Sucesivas
- Métodos de Linealización
  - Newton – Raphson
  - Modificado de Newton para Resolver Raíces Múltiples
  - Von Mises o Cuerdas Paralelas
  - Secante
  - Regula falsi y métodos relacionados.

Aproximaciones Sucesivas

Para utilizar este método necesitamos un punto y redefinir la ecuación de la siguiente manera:

Dada la ecuación original del tipo:

$$f(x) = 0$$

Debemos obtener una expresión en donde la incógnita es una función de si misma:

$$x = F(x)$$

Aproximaciones Sucesivas

Sea la función: $y = f(x)$

Se desea hallar x^* tal que $y = f(x^*) = 0$

Se debe definir una nueva función *F(x)* del tipo: $x = F(x)$

$F(x)$ {
- Se obtiene sumando miembro a miembro en la expresión original x
- Despejando x de algún termino

$f(x) = \sqrt{x} + \ln(x) - 4$ {
- $F(x) = \sqrt{x} + \ln(x) - 4 + x$
- $F(x) = (4 - \ln(x))^2$
- $F(x) = e^{(4-\sqrt{x})}$

Aproximaciones Sucesivas

A partir de un valor inicial (punto de arranque o valor semilla) se genera un nuevo valor utilizando *F(x)*

$$x^{(0)} = \alpha_0$$

$$x^{(1)} = F\left(x^{(0)}\right)$$

$$x^{(2)} = F\left(x^{(1)}\right)$$

$$x^{(k+1)} = F\left(x^{(k)}\right)$$

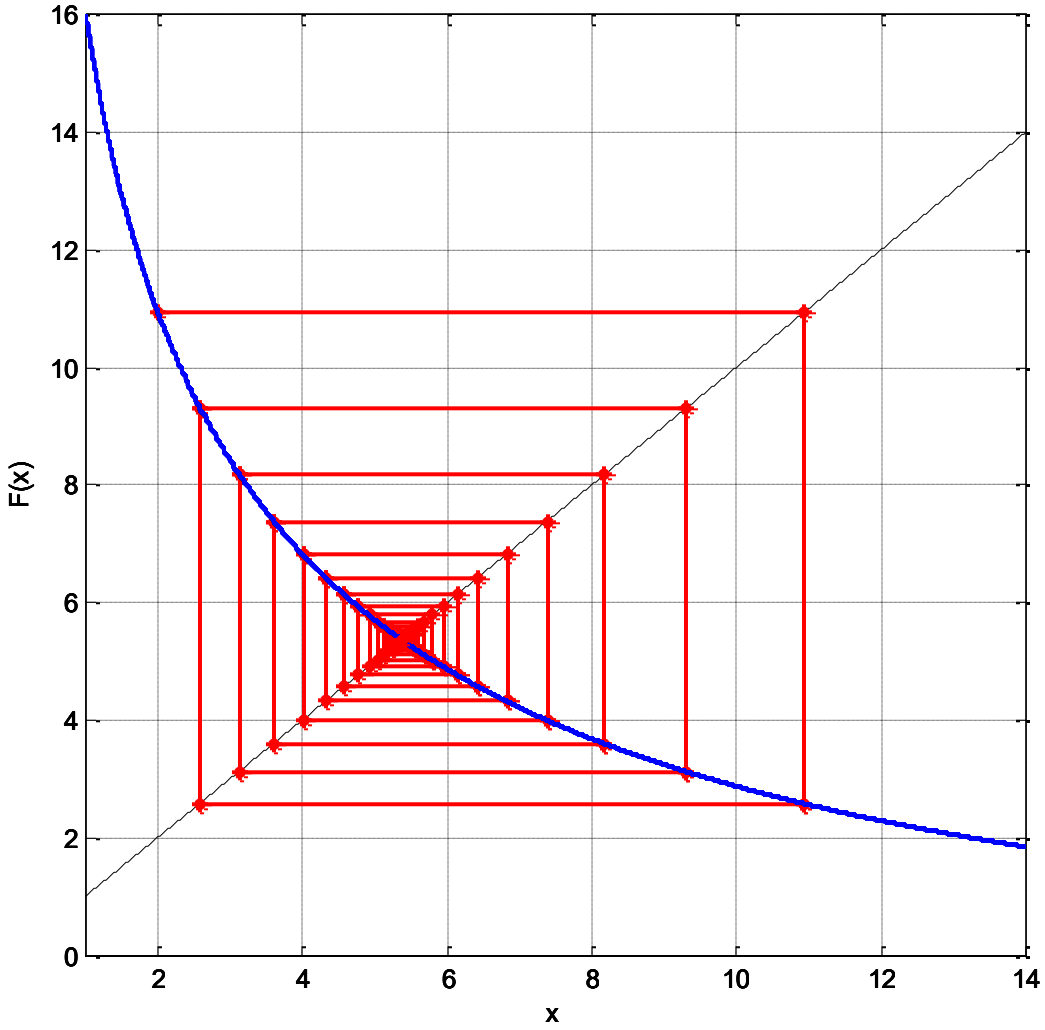
$$k = 1,\ 2,\ \ldots,\ k_{\max}$$

$$\left|\frac{x^{(k+1)} - x^{(k)}}{x^{(k)}}\right| < \varepsilon_r$$

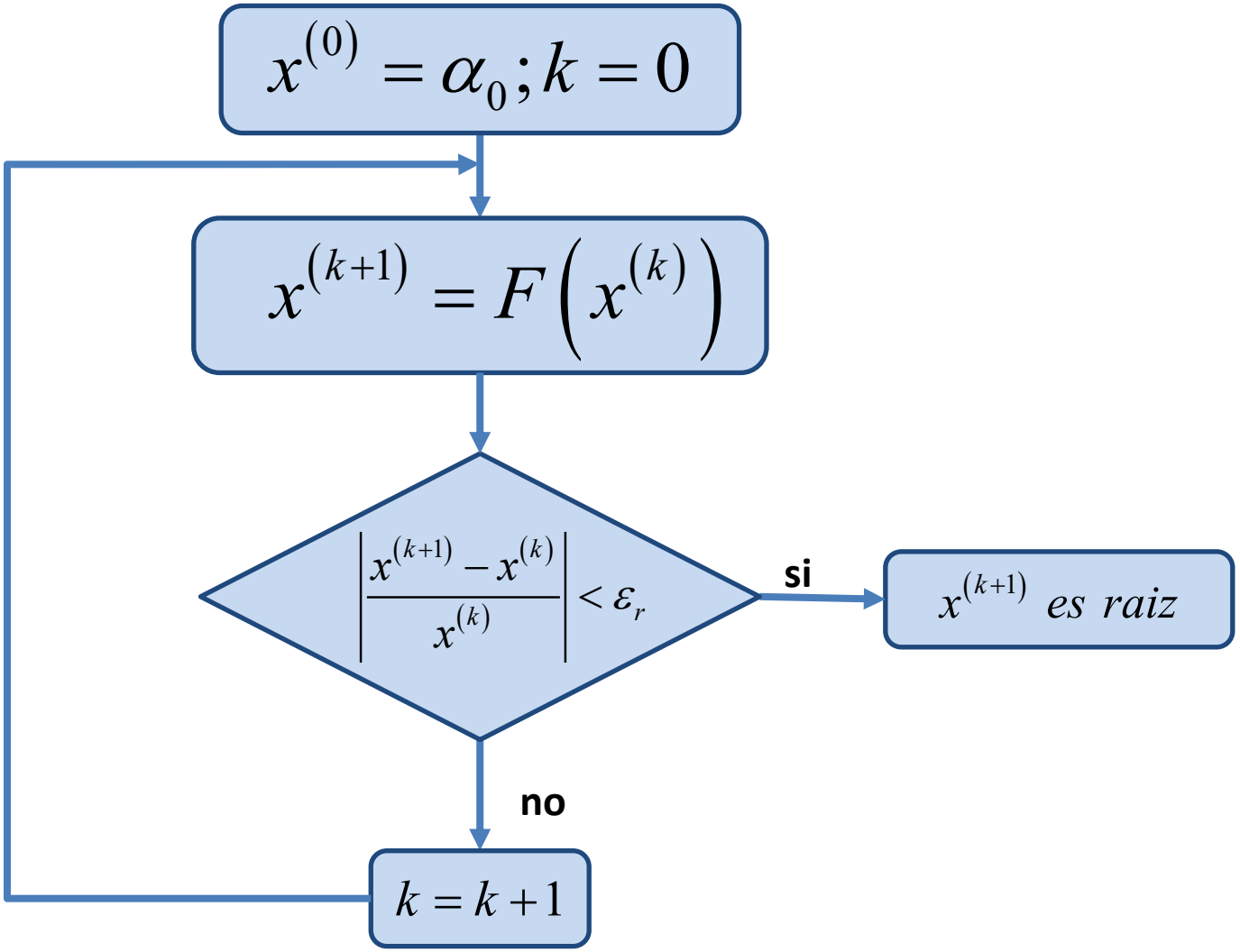
$$\left|x^{(k+1)} - F\left(x^{(k+1)}\right)\right| < \varepsilon$$

Se generan valores hasta satisfacer la tolerancia del error o alcanzar el numero máximo de iteraciones

Interpretación Grafica



Aproximaciones Sucesivas



Aproximaciones Sucesivas

i	x_i	$F(x_i)$	*Error*
0			
1			
2			
3			
4			
5			
6			

$$F(x) = e^{\left(4-\sqrt{x}\right)}$$

$$F(x) = \left(4 - \ln(x)\right)^2$$

$$F(x) = \sqrt{x} + \ln(x) - 4 + x$$

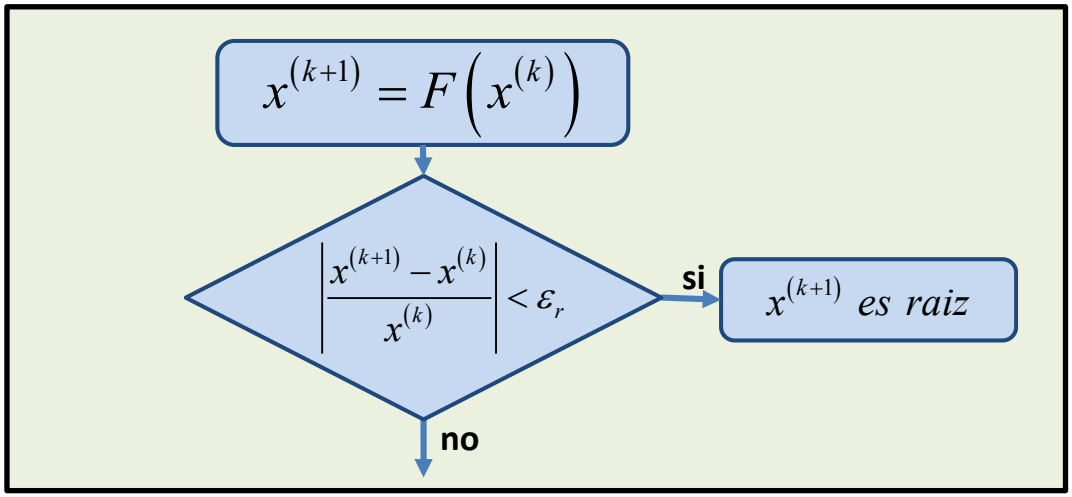
Aproximaciones Sucesivas (scilab)

```
function y=F(x)
y= sqrt(x) + log(x) – 4 + x;
endfunction

function y=F(x)
y= (4 - log(x)).^2;
endfunction

function y=F(x)
y= exp (4 – sqrt(x));
endfunction
```

Funciones de aproximación

Aproximaciones Sucesivas (scilab)

```
function out=aproxsuc(fun, x0, tol)
x=x0;
for k=1:100
```




```
end
if k == 100
        out=[];
        disp('no converge');
end
endfunction
```

Método de Wegstein

Mejora el proceso de sustitución directa.
A partir del valor inicial se generan un valor de manera tradicional y luego se aplica la siguiente ley recursiva:

$$\omega = \frac{F\left(x^{(k)}\right) - F\left(x^{(k-1)}\right)}{x^{(k)} - x^{(k-1)}}$$

$$q = \frac{\omega}{\omega - 1}$$

$$x^{(k+1)} = qx^{(k)} + \left(1 - q\right) F\left(x^{(k)}\right)$$

$$k = 1, 2, 3, \ldots, k_{\max}$$

$$\left| \frac{x^{(k+1)} - x^{(k)}}{x^{(k)}} \right| < \varepsilon_r$$

$$\left| x^{(k+1)} - F\left(x^{(k+1)}\right) \right| < \varepsilon$$

Se generan valores hasta satisfacer la tolerancia del error o alcanzar el numero máximo de iteraciones

Método de Wegstein

Paso a paso para una mejor comprensión:

$$x^{(0)} = \alpha_0$$

$$x^{(1)} = F\left(x^{(0)}\right)$$

$$\omega = \frac{F\left(x^{(1)}\right) - F\left(x^{(0)}\right)}{x^{(1)} - x^{(0)}}$$

$$q = \frac{\omega}{\omega - 1}$$

$$x^{(2)} = qx^{(1)} + (1-q)\,F\left(x^{(1)}\right)$$

$$\omega = \frac{F\left(x^{(k)}\right) - F\left(x^{(k-1)}\right)}{x^{(k)} - x^{(k-1)}}$$

$$q = \frac{\omega}{\omega - 1}$$

$$x^{(k+1)} = qx^{(k)} + (1-q)\,F\left(x^{(k)}\right)$$

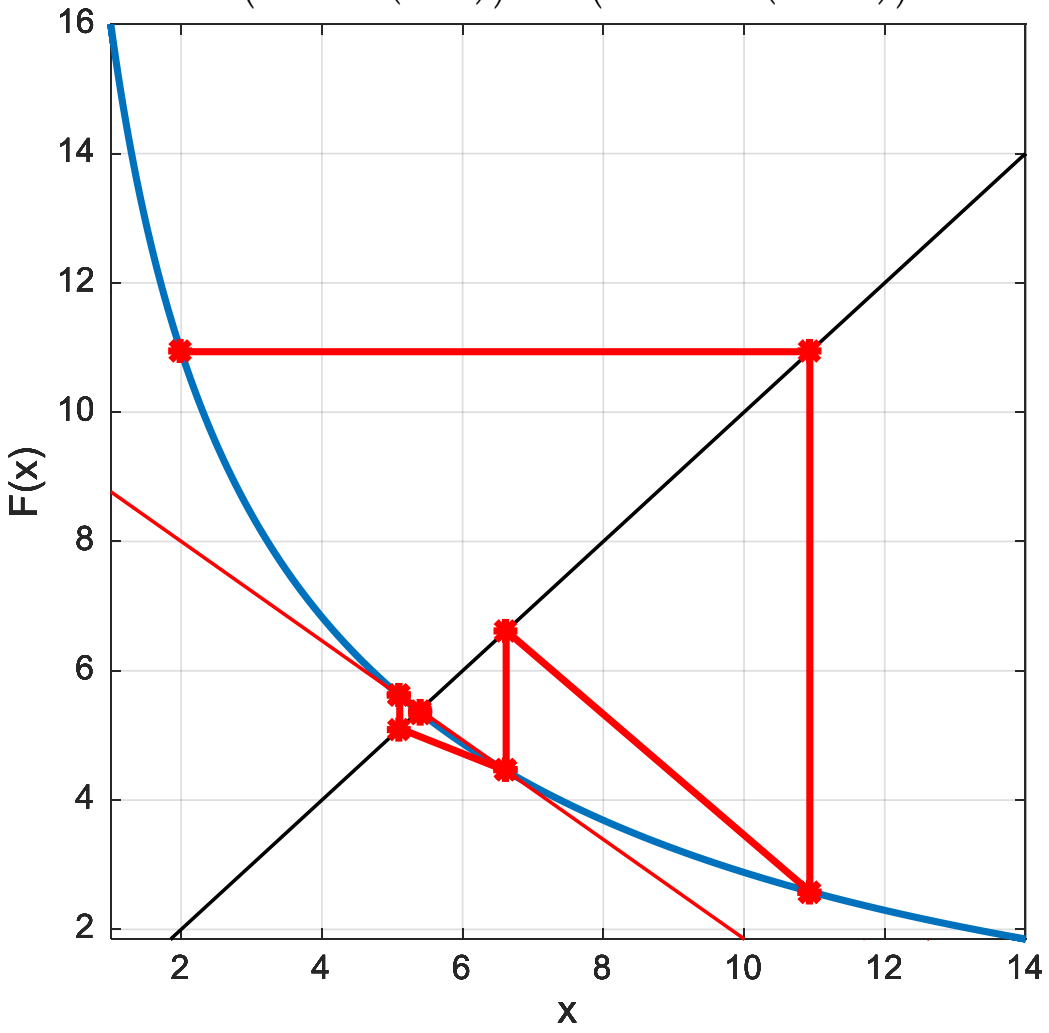
$$k = 1, 2, 3, \ldots, k_{\max}$$

$$\left|\frac{x^{(k+1)} - x^{(k)}}{x^{(k)}}\right| < \varepsilon_r$$

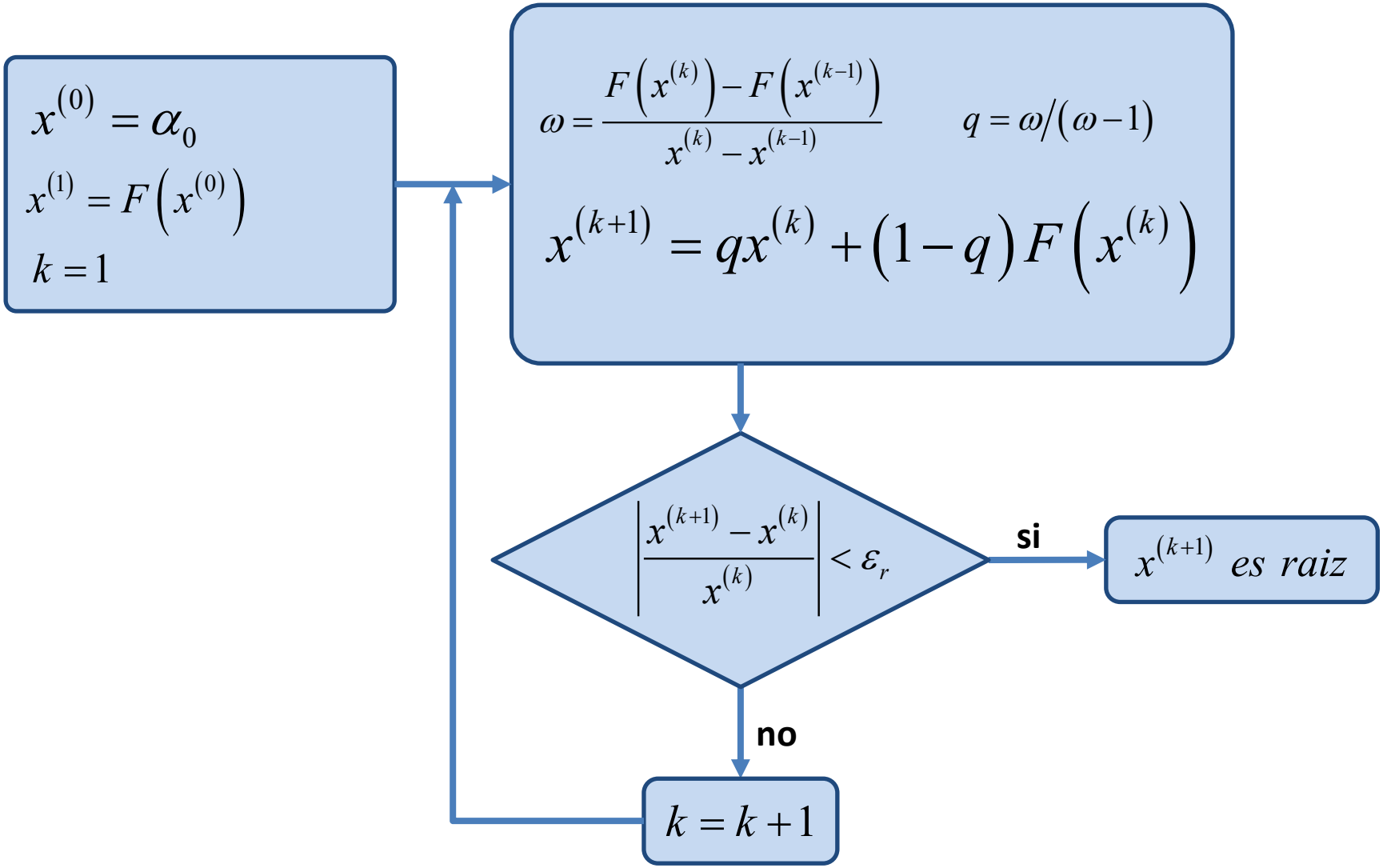
$$\left|x^{(k+1)} - F\left(x^{(k+1)}\right)\right| < \varepsilon$$

Interpretación Grafica

El nuevo valor generado en la iteración k, o sea $x^{(k+1)}$ corresponde a la intersección de la recta que une los puntos $\left(x^{(k)}, F\left(x^{(k)}\right)\right)$ y $\left(x^{(k-1)}, F\left(x^{(k-1)}\right)\right)$ con la recta $y=x$



Método de Wegstein

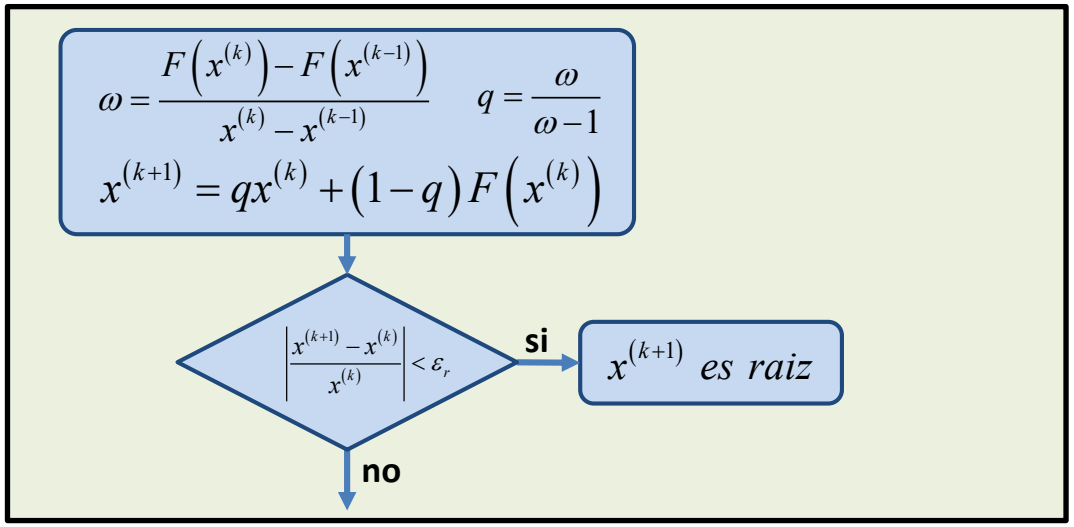


Método de Wegstein

i	x_i	$F(x_i)$	ω	*q*	*Error*
0					
1					
2					
3					
4					
5					
6					

Método de Wegstein (scilab)

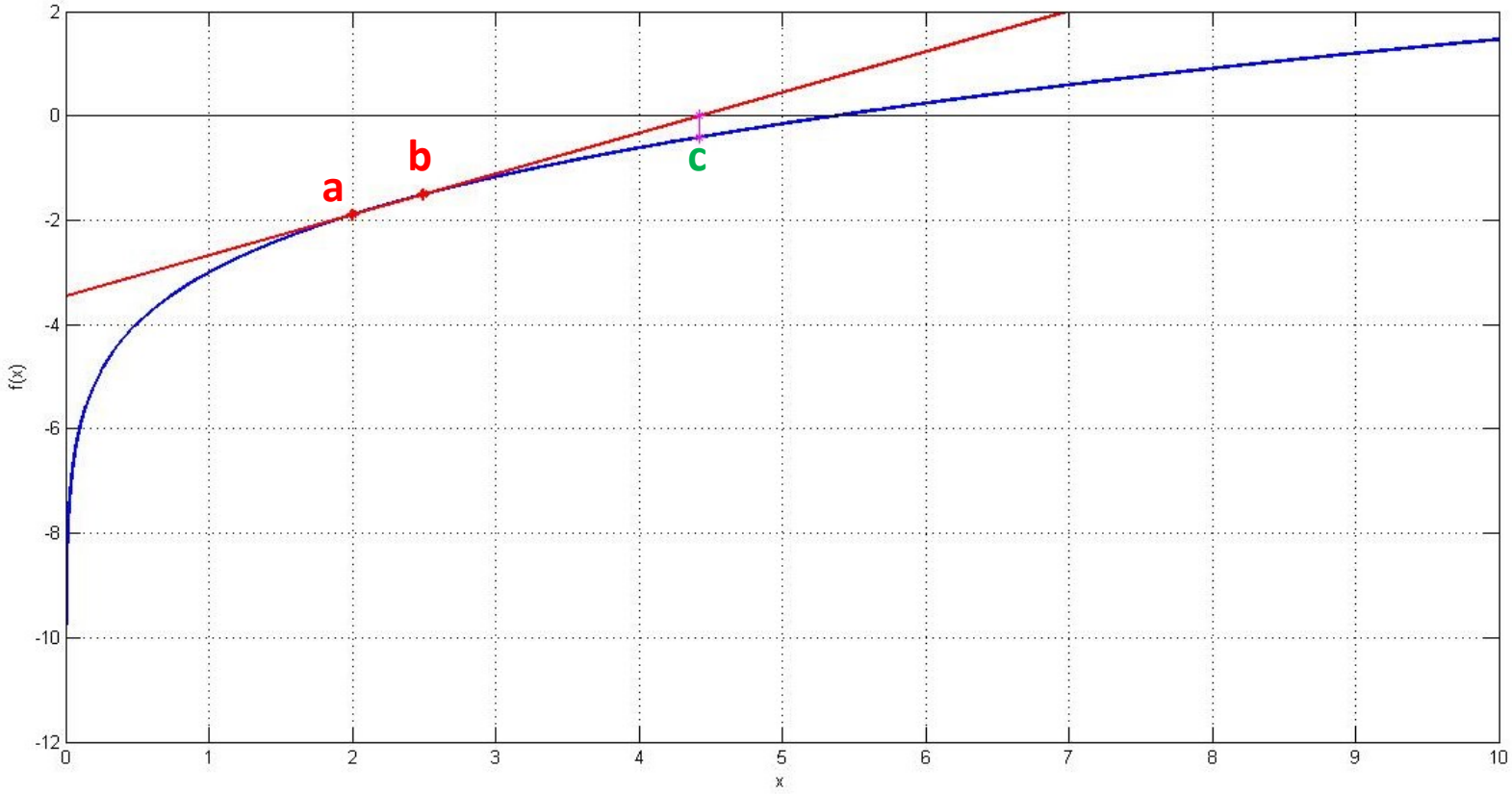
```
function out=wegstein(fun, x0, tol)
x(1)=x0;
x(2)=fun(x(1));
FF=fun(x);
for k=2:100
```



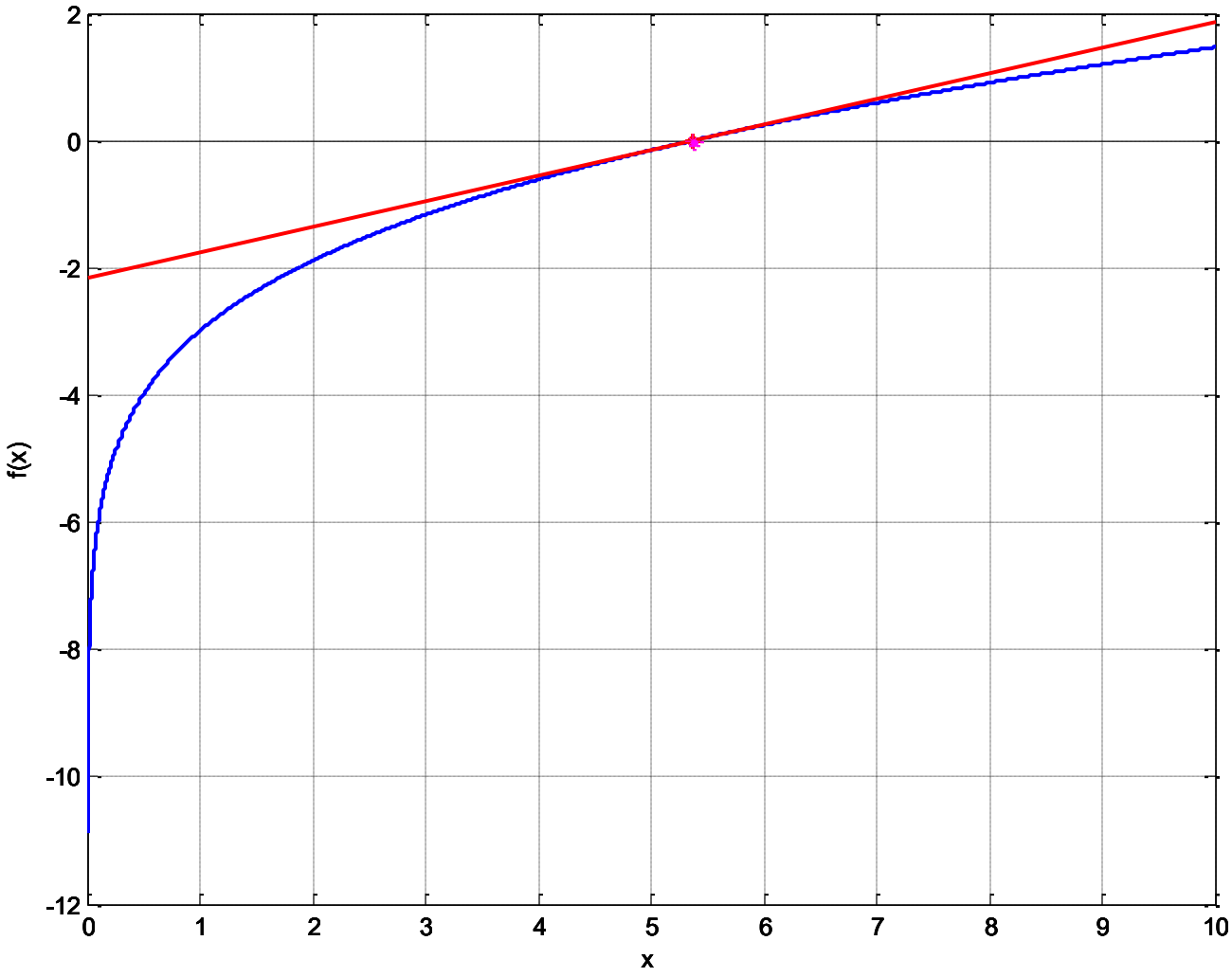

```
end
if k == 100
        out=[];
        disp('no converge');
end
endfunction
```

Método de la Secante

Para utilizar este método solo necesitamos dos puntos y la recta que los une. El nuevo punto surge de su intersección con el eje de las abscisas.



Método de la Secante



Método de la secante

A partir de dos valores iniciales (puntos de arranque) se genera un nuevo valor que corresponde a la intersección de la secante a la curva con el eje de las abscisas.

$$x^{(0)} = \alpha_0$$

$$x^{(1)} = \alpha_1$$

$$x^{(2)} = x^{(1)} - \frac{f\left(x^{(1)}\right)\left(x^{(1)} - x^{(0)}\right)}{f\left(x^{(1)}\right) - f\left(x^{(0)}\right)}$$

$$x^{(3)} = x^{(2)} - \frac{f\left(x^{(2)}\right)\left(x^{(2)} - x^{(1)}\right)}{f\left(x^{(2)}\right) - f\left(x^{(1)}\right)}$$

$$x^{(k+1)} = x^{(k)} - \frac{f\left(x^{(k)}\right)\left(x^{(k)} - x^{(k-1)}\right)}{f\left(x^{(k)}\right) - f\left(x^{(k-1)}\right)}$$

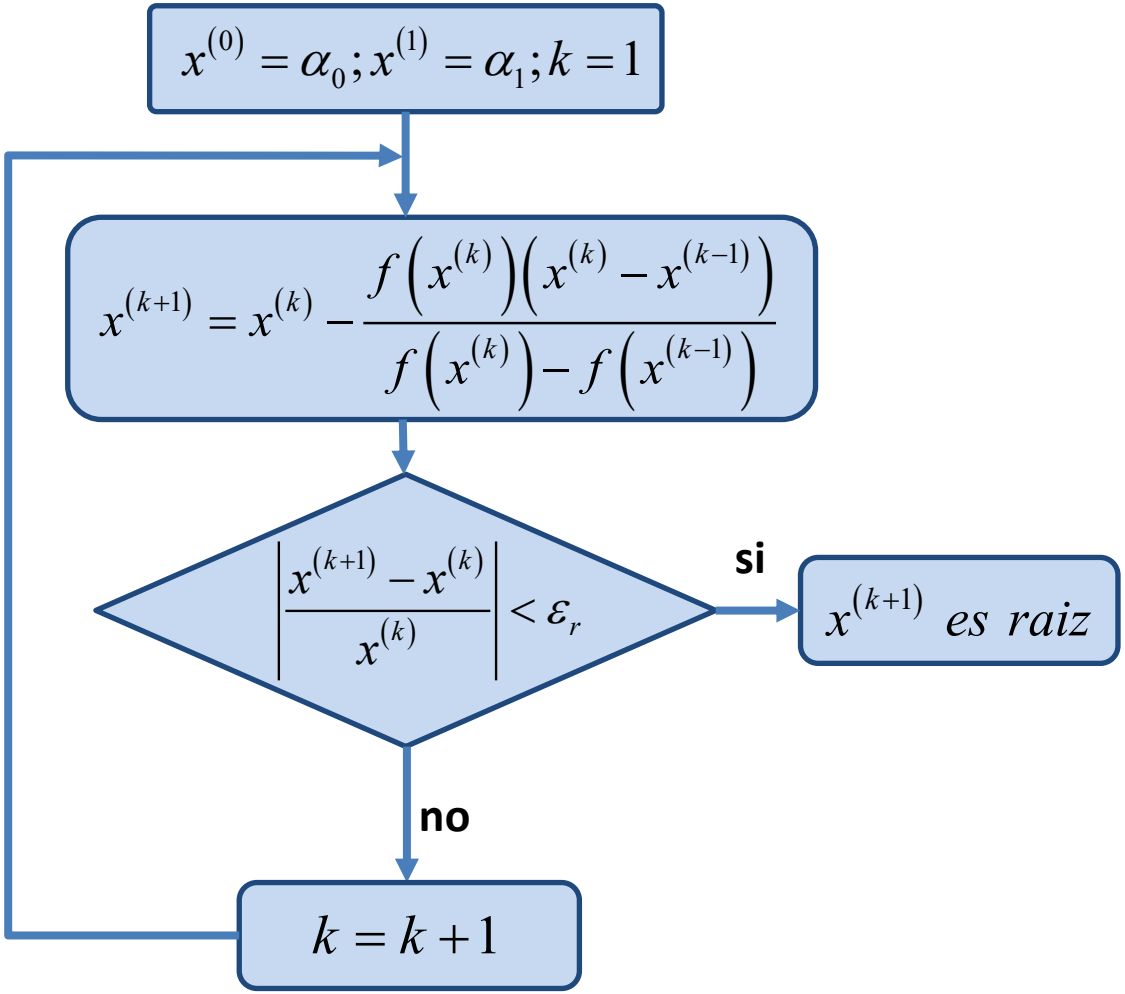
$$k = 1,\ 2,\ ...,\ k_{\max}$$

$$\left|\frac{x^{(k+1)} - x^{(k)}}{x^{(k)}}\right| < \varepsilon_r$$

$$\left|f\left(x^{(k+1)}\right)\right| < \varepsilon$$

Se generan valores hasta satisfacer la tolerancia del error o alcanzar el numero máximo de iteraciones

Método de la Secante



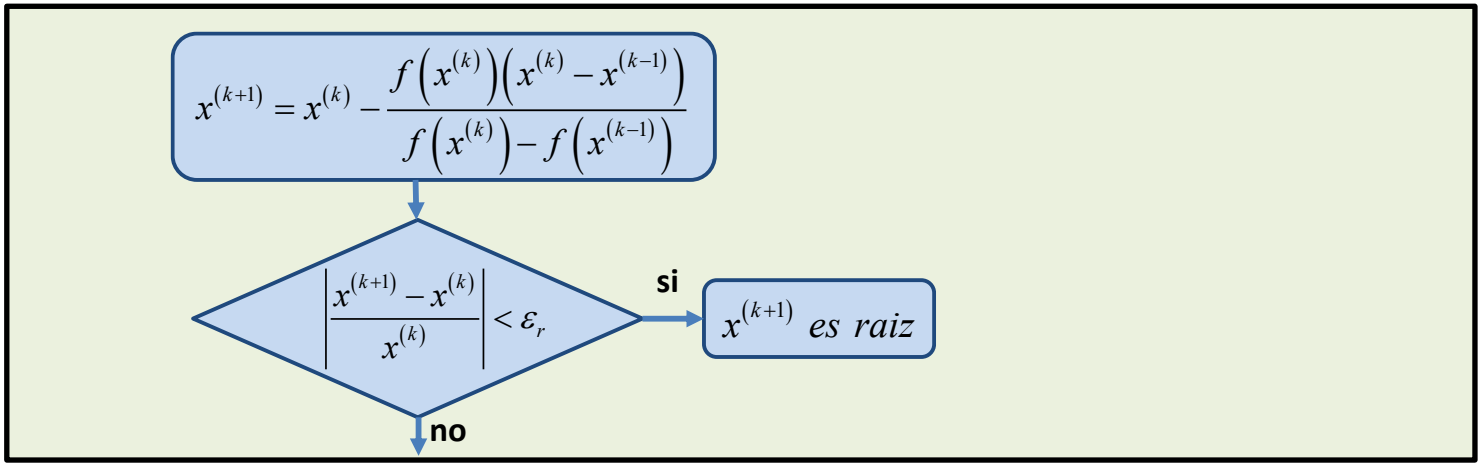
Método de la Secante

i	x_i	$f(x_i)$	*Error*
0			
1			
2			
3			
4			
5			
6			

$$f(x) = \ln(x) + e^{2x}$$

Método de la Secante (scilab)

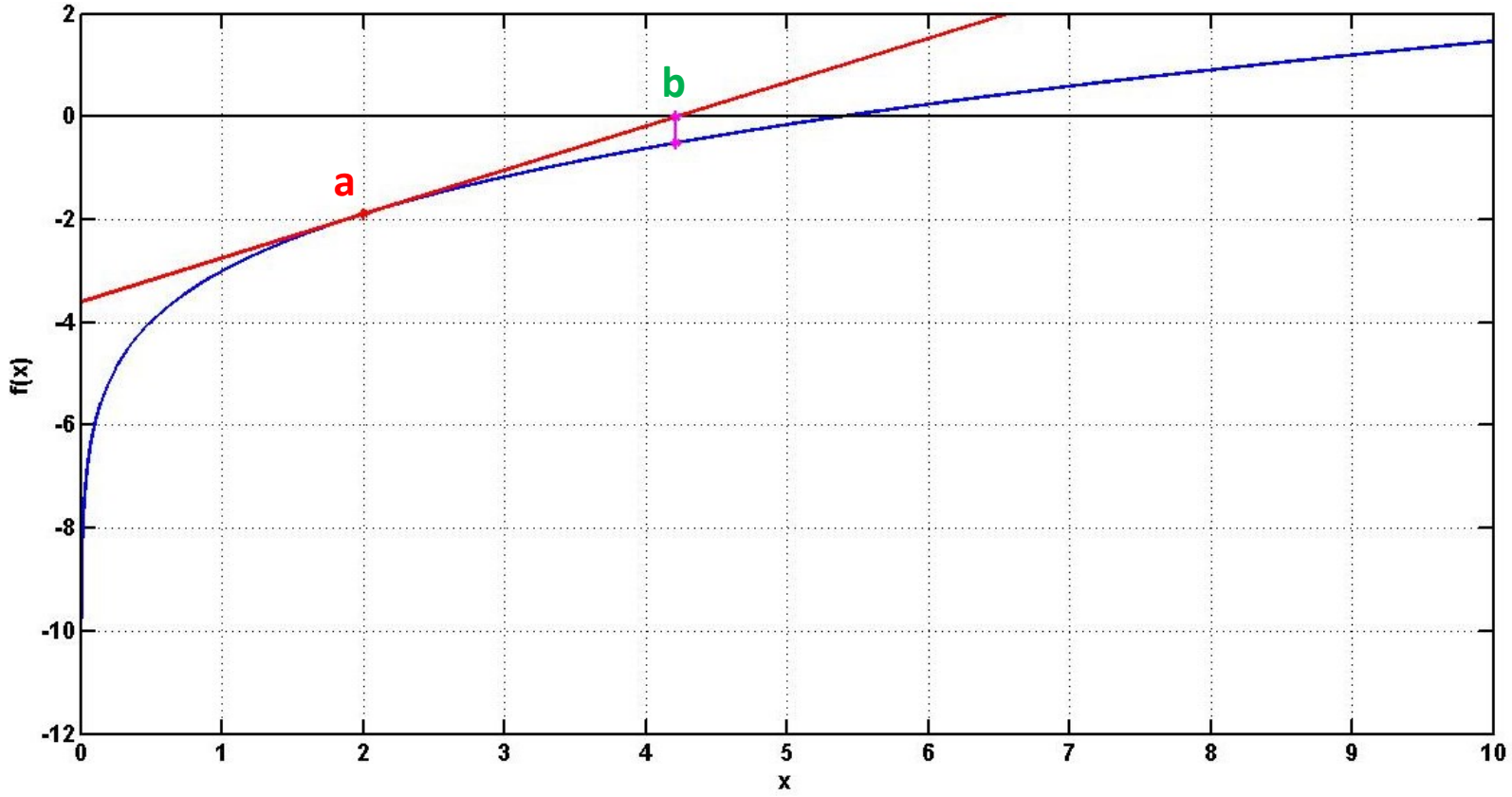
```
function out=secante(fun, x0, x1, tol)
x(1)=x0;
x(2)=x1;
for k=2:100
```




```
end
if k == 100
        out=[];
        disp('no converge');
end
endfunction
```

Método de Newton-Raphson

Para utilizar este método solo necesitamos un punto y la derivada de la función. El nuevo punto surge de la intersección de la recta tangente con el eje de las abscisas.



Método de Newton-Raphson

A partir de un valor inicial (punto de arranque o valor semilla) se genera un nuevo valor que corresponde a la intersección de la recta tangente a la curva con el eje de las abscisas.

$$x^{(0)} = \alpha_0$$

$$x^{(1)} = x^{(0)} - \frac{f\left(x^{(0)}\right)}{f'\left(x^{(0)}\right)}$$

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

$$x^{(k+1)} = x^{(k)} - \frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}$$

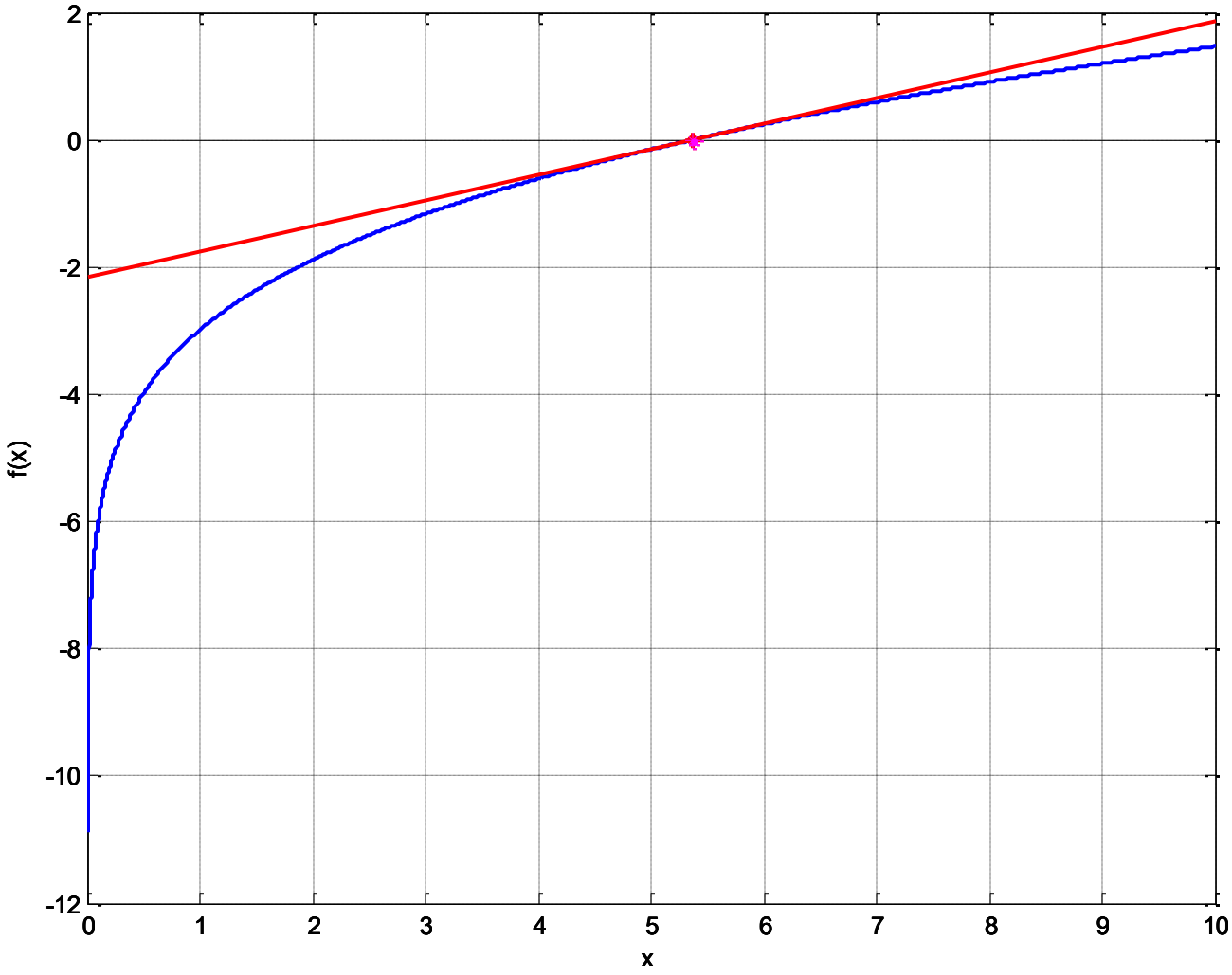
$$k = 1,\ 2,\ \ldots,\ k_{\max}$$

$$\left|\frac{x^{(k+1)} - x^{(k)}}{x^{(k)}}\right| < \varepsilon_r$$

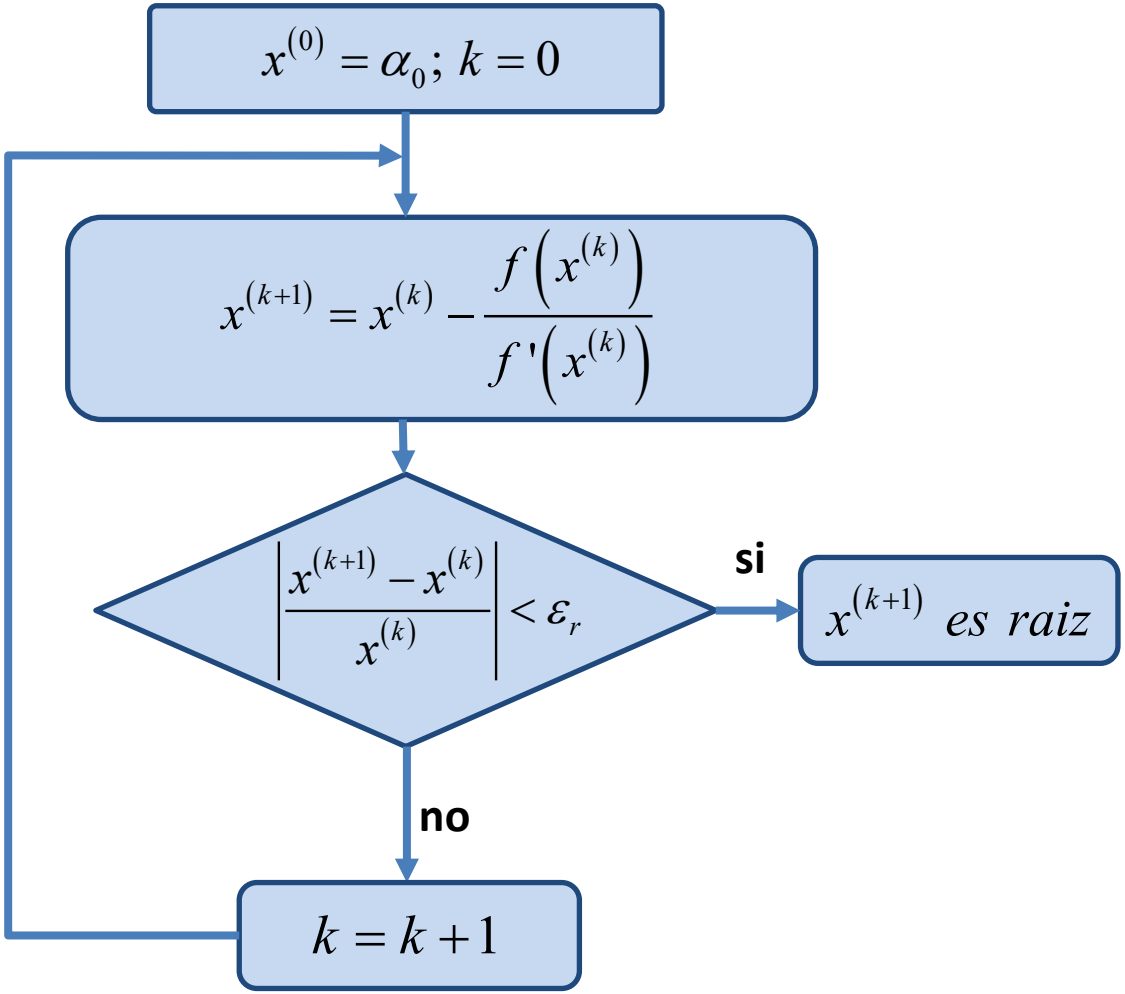
$$\left|f\left(x^{(k+1)}\right)\right| < \varepsilon$$

Se generan valores hasta satisfacer la tolerancia del error o alcanzar el numero máximo de iteraciones

Método de Newton-Raphson



Método de Newton-Raphson



Método de Newton-Raphson

i	x_i	$f(x_i)$	$f'(x_i)$	*Error*
0				
1				
2				
3				
4				
5				
6				

Método de Newton-Raphson (scilab)

```
function y=f(x)
y=sqrt(x) + log(x) - 4;
endfunction
```

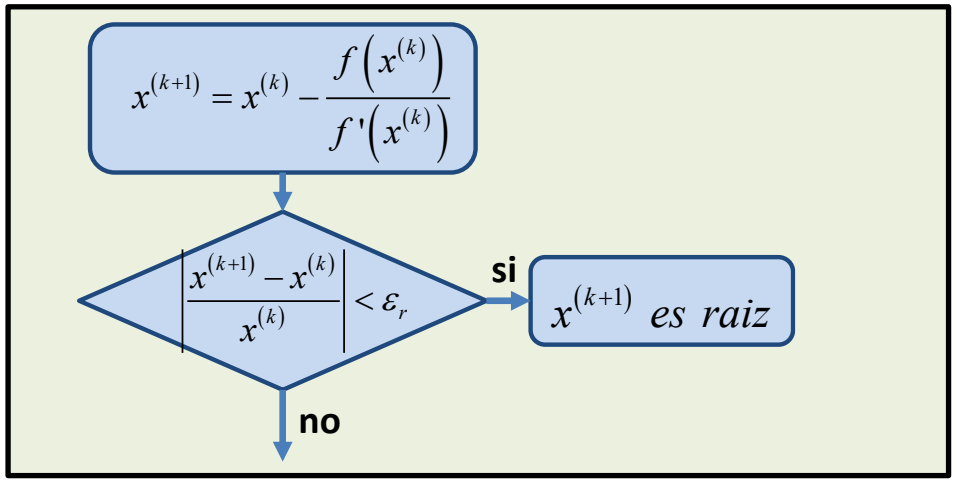
Función

```
function y=fp(x)
y=0.5./sqrt(x) + 1./(x);
endfunction
```

Derivada

Método de Newton-Raphson (scilab)

```
function out=newtonR(fun, funp, x0, tol)
x=x0;
for k=1:100
```




```
end
if k == 100
        out=[];
        disp('no converge');
end
endfunction
```

Modificación de N-R

$$u(x) = \frac{f(x)}{f'(x)}$$

Ralston and Rabinowitz (1978) propusieron la función *u(x)* que tiene las mismas raíces que *f(x)*

Por lo tanto, al aplicar N-R a la nueva función *u(x)* se obtiene: $x^{(k+1)} = x^{(k)} - \frac{u\left(x^{(k)}\right)}{u'\left(x^{(k)}\right)}$

luego,

$$\frac{u(x)}{u'(x)} = \frac{f(x)f'(x)}{\left[f'(x)\right]^2 - f(x)f''(x)}$$

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

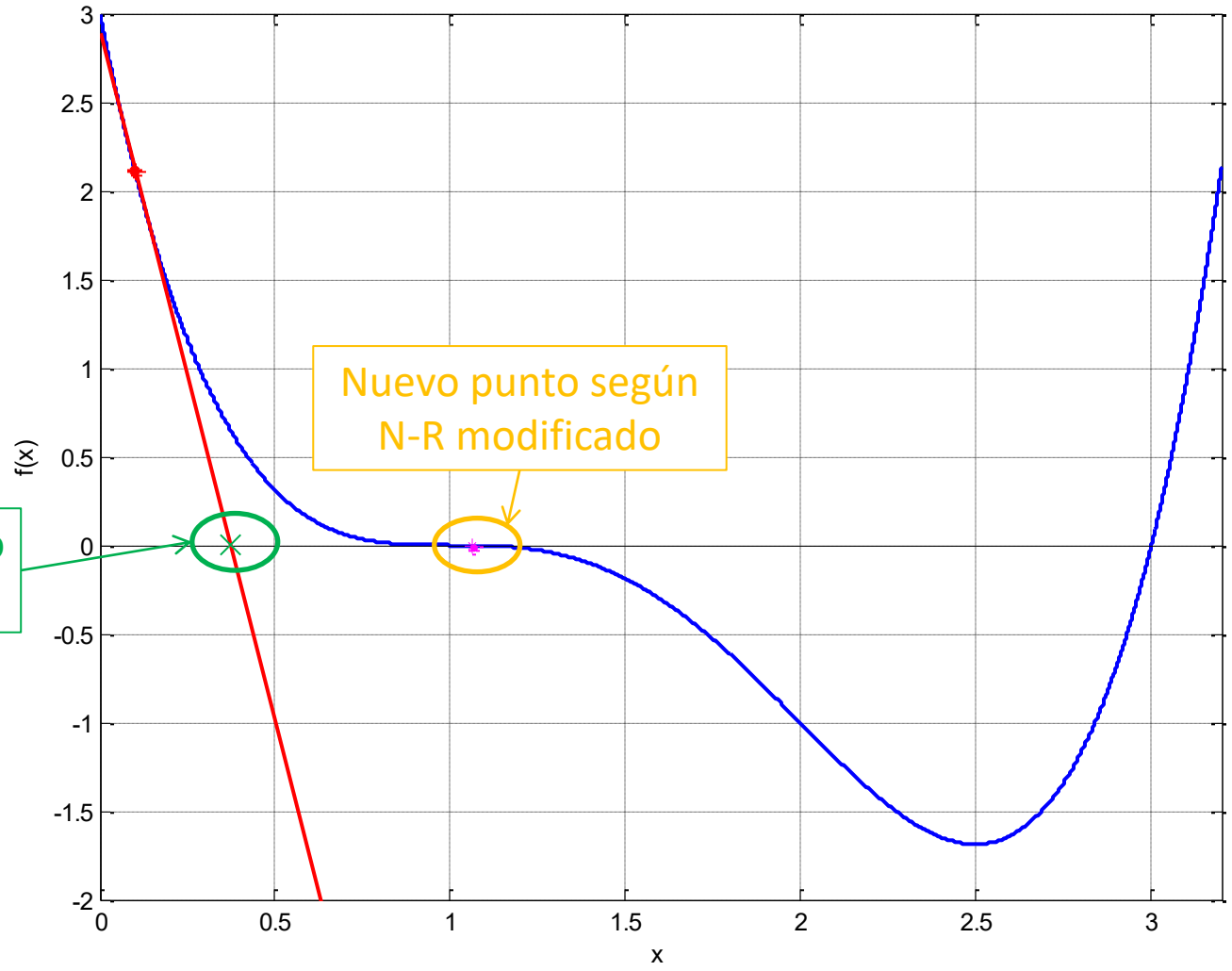
$$k = 1,\ 2,\ \ldots,\ k_{\max}$$

$$\left|\frac{x^{(k+1)} - x^{(k)}}{x^{(k)}}\right| < \varepsilon_r$$

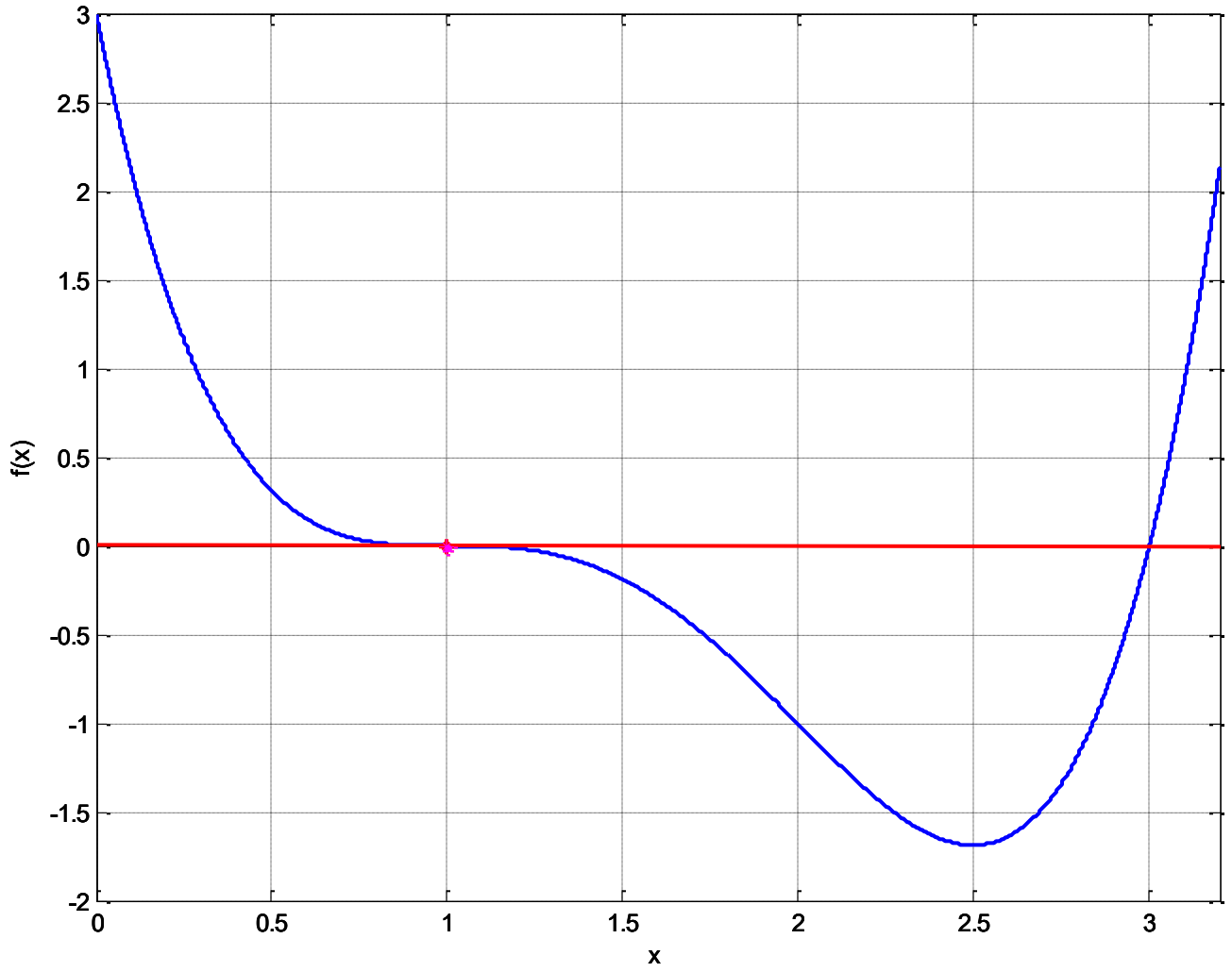
$$\left|f\left(x^{(k+1)}\right)\right| < \varepsilon$$

Se generan valores hasta satisfacer la tolerancia del error o alcanzar el numero máximo de iteraciones

Interpretación Grafica

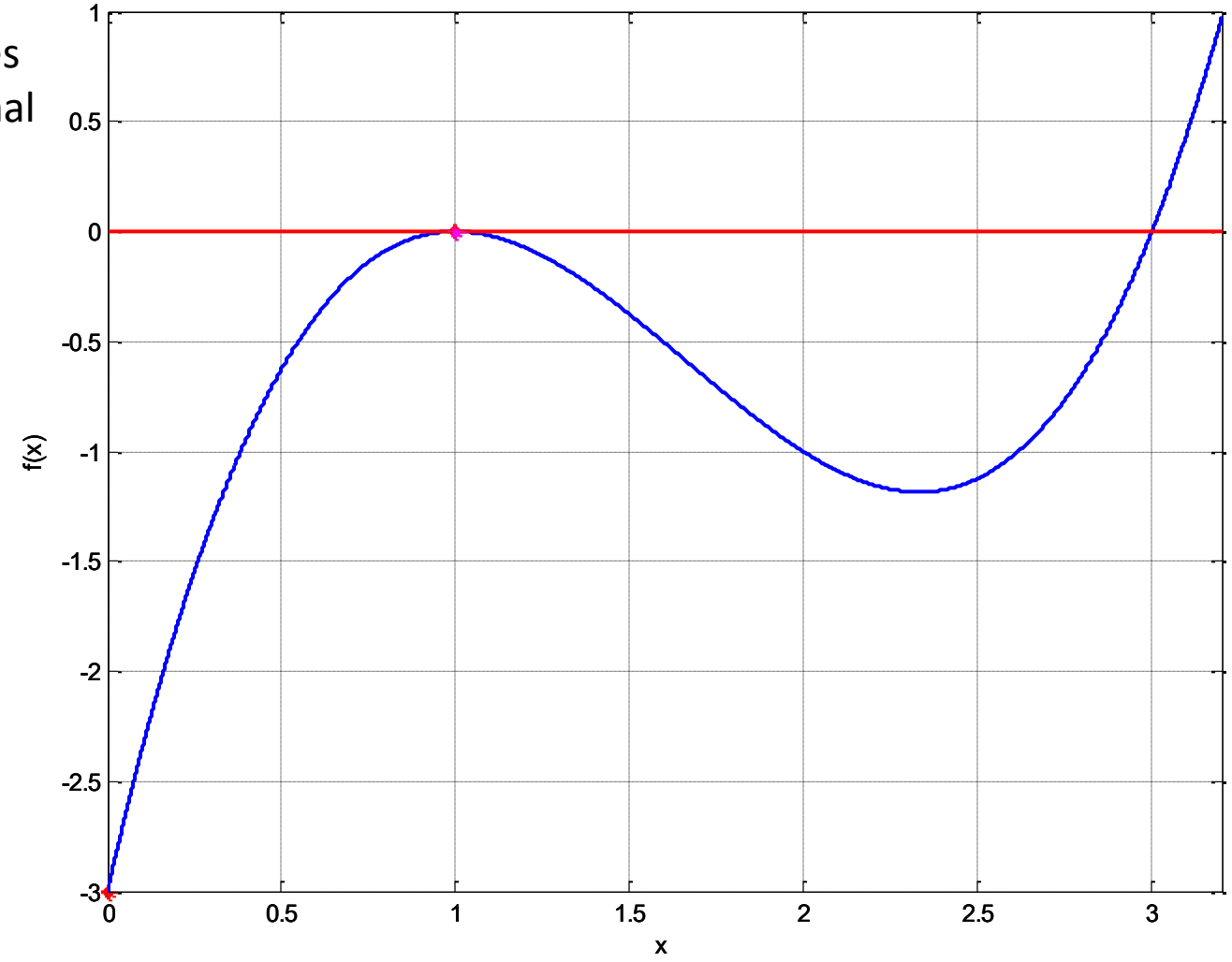


Interpretación Grafica



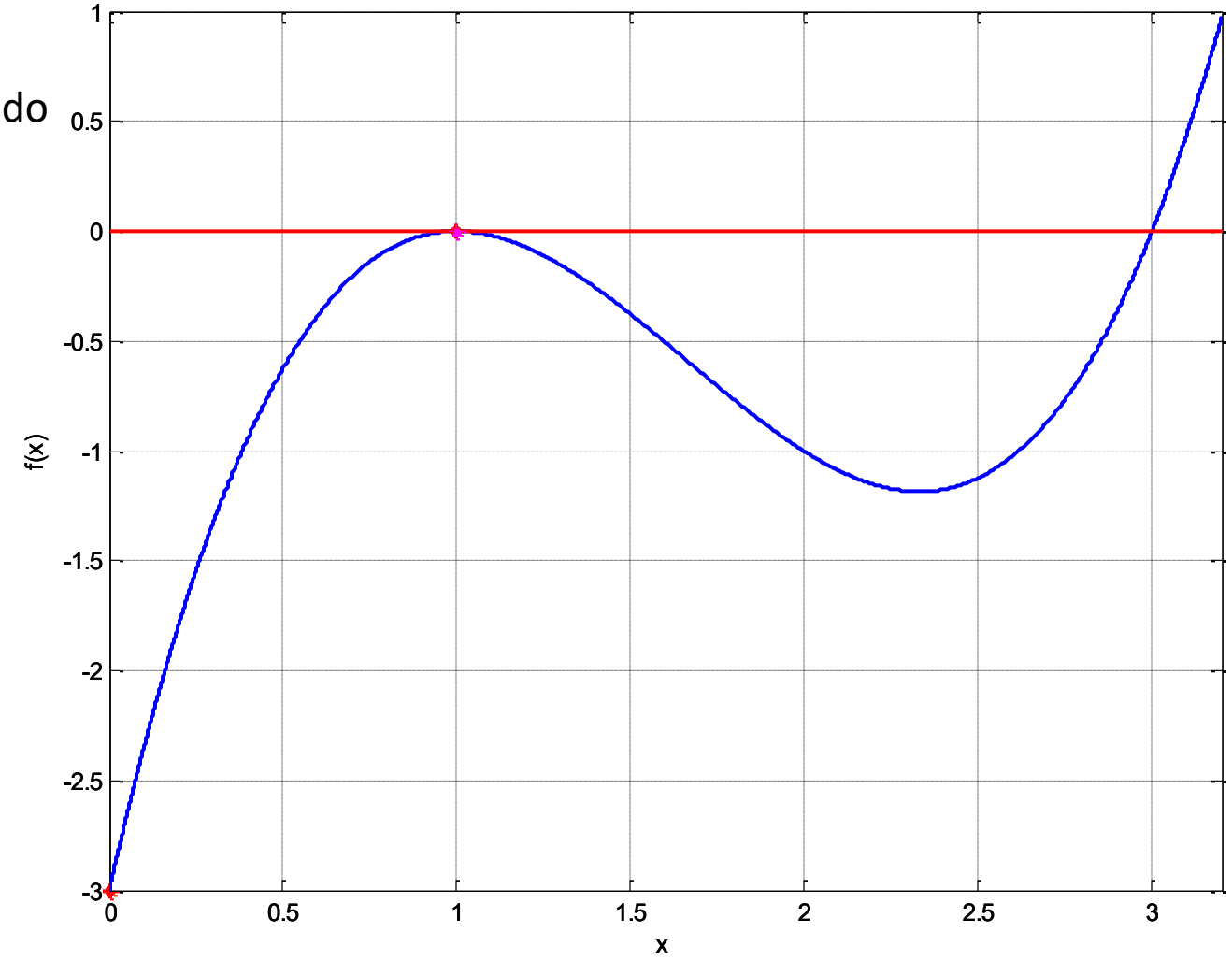
Interpretación Grafica (N-R tradicional)

24 iteraciones
N-R tradicional

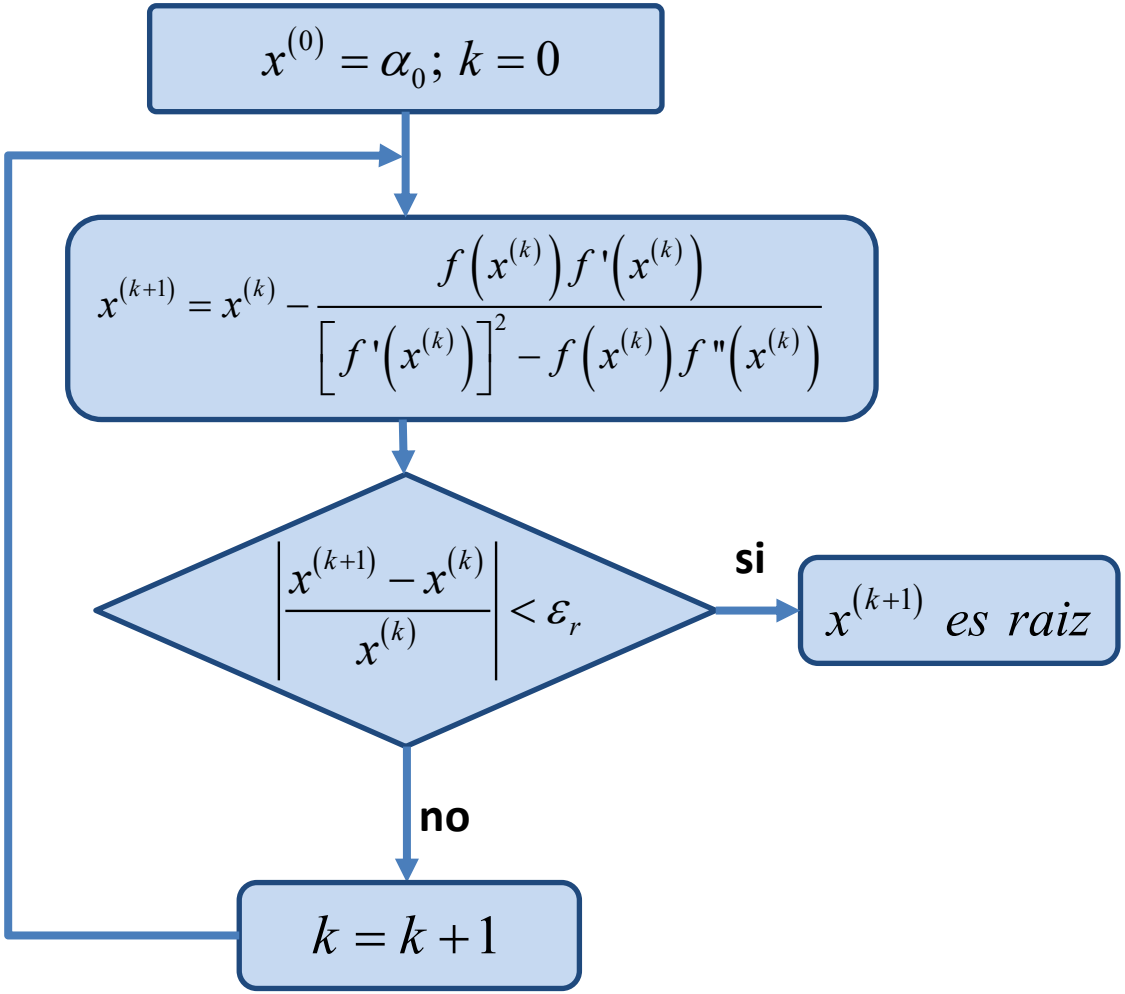


Interpretación Grafica (N-R modificado)

5 iteraciones
N-R modificado



Método de N-R modificado



Método de N-R modificado

i	x_i	$f(x_i)$	$f'(x_i)$	$f''(x_i)$	*Error*
0					
1					
2					
3					
4					
5					
6					

N-R modificado (scilab)

Modificar el código de Newton-Rhapson y adaptarlo a este método

Técnica para resolver ecuaciones cúbicas de estado

- Cualquier polinomio cúbico (incluida las EoS) puede expresarse de la siguiente forma:

$$z^3 + az^2 + bz + c = 0$$

- La primera raíz se encuentra utilizando la modificación de N-R modificado. Se sugiere utilizar como valor semilla el factor de compresibilidad de los gases ideales $z^{(0)}=1$

- Se utiliza la primera raíz hallada para factorizar el polinomio:

$$\left(z^3 + az^2 + bz + c\right) \Big| \underline{z - \alpha_1}$$
$$Q(z)$$

- Debido a la factorización se cumple que:

$$z^3 + az^2 + bz + c = \left(z - \alpha_1\right) Q\left(z\right)$$

- Ambos miembros tienen las mismas raíces por lo que las dos restante corresponden a las del polinomio $Q(z)$

- ¡$Q(z)$ es una cuadrática!

Técnica para resolver ecuaciones cúbicas de estado

- Factorización del polinomio original: $\left(z^3 + az^2 + bz + c\right) \Big| \underline{z - \alpha_1}$ → $Q(z)$

	1	a	b	c
α_1		α_1	$\alpha_1\left(\alpha_1 + a\right)$	$\alpha_1\left(b + \alpha_1\left(\alpha_1 + a\right)\right)$
	1	$\alpha_1 + a$	$b + \alpha_1\left(\alpha_1 + a\right)$	$c + \alpha_1\left(b + \alpha_1\left(\alpha_1 + a\right)\right)$

0 (cero)

$$Q(z) = z^2 + \left(\alpha_1 + a\right)z + \left(b + \alpha_1\left(\alpha_1 + a\right)\right)$$

$$D = \left(\alpha_1 + a\right)^2 - 4\left(b + \alpha_1\left(\alpha_1 + a\right)\right)$$

- $D \geq 0$ Las raíces restantes son reales
- $D<0$ Las raíces restantes son imaginarias ($D<0$)

Técnica para resolver ecuaciones cúbicas (Resumen)

$$z^3 + az^2 + bz + c = 0$$

1. La primera raíz se encuentra utilizando la modificación de N-R modificado. Se sugiere utilizar como valor semilla $z^{(0)}=1$

$$z^{(k+1)} = z^{(k)} - \frac{f\left(z^{(k)}\right) f'\left(z^{(k)}\right)}{\left[f'\left(z^{(k)}\right)\right]^2 - f\left(z^{(k)}\right) f''\left(z^{(k)}\right)}$$

2. Se analiza el discriminante de la nueva ecuación cuadrática

$$D = \left(\alpha_1 + a\right)^2 - 4\left(b + \alpha_1\left(\alpha_1 + a\right)\right)$$

$$D \geq 0 \begin{cases} \alpha_2 = \dfrac{-\left(\alpha_1 + a\right) + \sqrt{D}}{2} \\ \alpha_3 = \dfrac{-\left(\alpha_1 + a\right) - \sqrt{D}}{2} \end{cases} \Rightarrow \begin{aligned} z_v &= max\left(\alpha_1, \alpha_2, \alpha_3\right) \\ z_l &= min\left(\alpha_1, \alpha_2, \alpha_3\right) \end{aligned}$$

$D<0$ Las restantes raíces no son de interés físico

Crear la function siguiendo los pasos de la pagina anterior

Ejemplo: Calculo de la densidad del propano

Se desea calcular la densidad de la fase gaseosa del propano a 300 K y 9.98316 bar utilizando Peng Robinson.

$$T_c = 369.89801\ K;\ P_c = 42.56660\ bar;\ \omega = 0.15240;\ R = 83.14472\ \frac{cm^3\ bar}{mol\ K}$$

$$a = \frac{0.45724 R^2 T_c^{\,2}}{P_c} \qquad T_r = \frac{T}{T_c}$$

$$b = \frac{0.07780 R T_c}{P_c} \qquad \alpha(T_r) = \left[1 + \left(0.37464 + 1.54226\omega - 0.2699\omega^2\right)\left(1 - T_r^{1/2}\right)\right]^2$$

$$B' = \frac{bP}{RT} \qquad \Theta' = \frac{a\alpha(T_r)P}{(RT)^2}$$

$$z^3 + \left(B' - 1\right)z^2 + \left(\Theta' - 3B'^2 - 2B'\right)z + \left(B'^3 + B'^2 - \Theta' B'\right) = 0$$

$$v_v = z_v RT / P$$