

Unidad 2: Solución de ecuaciones

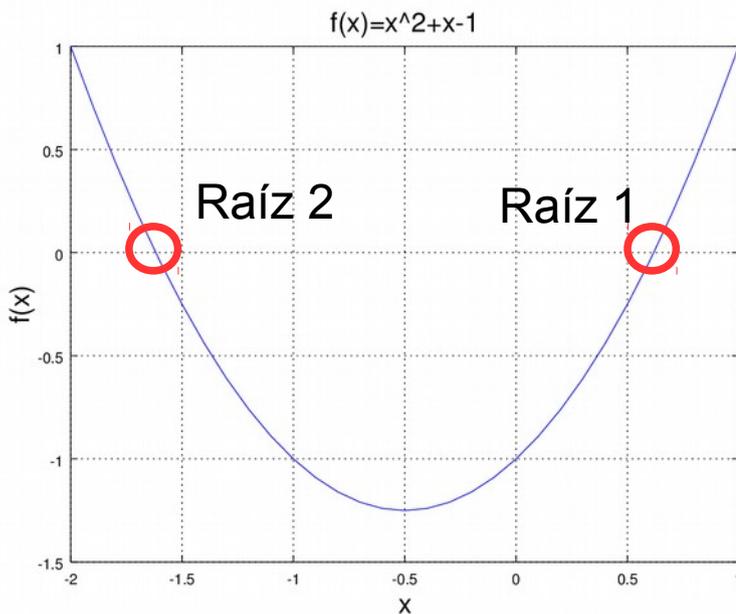
Temario:

- Raíces de funciones
- Método de la Bisección
- Método de Regula Falsi
- Método de Newton-Raphson
- Método de la Secante
- Múltiples raíces

Raíces (soluciones) de funciones

$$f(x) = x^2 + x - 1 = 0$$

$$x = \frac{-1 \pm \sqrt{1^2 - 4(-1)}}{2} \rightarrow \begin{cases} x_1 = -\frac{1}{2} + \sqrt{\frac{5}{4}} = 0,61803398874989484820458683436564 \\ x_2 = -\frac{1}{2} - \sqrt{\frac{5}{4}} = -1,6180339887498948482045868343656 \end{cases}$$



x_1 y x_2 son las “raíces” o “ceros” de la ecuación cuadrática. Es decir son los **valores de x que cumplen la ecuación.**

Para encontrar los ceros se puede:

- Hacer “prueba y error” evaluando la función un número de veces que dependerá de la suerte y la experiencia.
- Graficar la función, evaluando la función en (muchísimos) puntos y estimar un intervalo.
- Usar métodos numéricos basados en algoritmos lógicos.

Métodos numéricos

Hay dos clases de métodos numéricos, los cerrados y los abiertos.

En los **métodos cerrados** se necesita **conocer un entorno** que contiene la raíz. Se comienza con un **intervalo cerrado de partida $[a,b]$** en el que **$f(a)$ y $f(b)$ tengan distinto signo** y por el **teorema de Bolzano** queda asegurado que existe una solución. El algoritmo consiste en **ir reduciendo iterativamente el ancho del intervalo** hasta que se obtenga un **error aceptable**. Los métodos cerrados que veremos son el método de la bisección y el de regula falsi.

En los **métodos abiertos** se comienza por **un punto de partida** y se predice una mejor aproximación de la solución utilizando el valor de la derivada de la función. Los métodos abiertos que veremos son el método de Newton-Raphson y el de la secante.

Algoritmo de la bisección

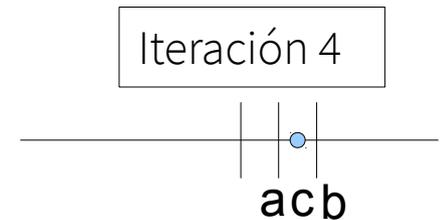
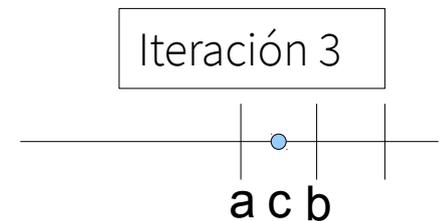
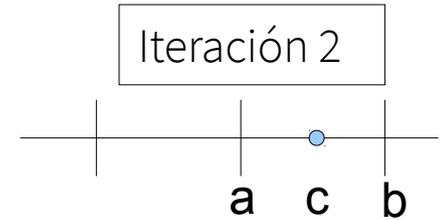
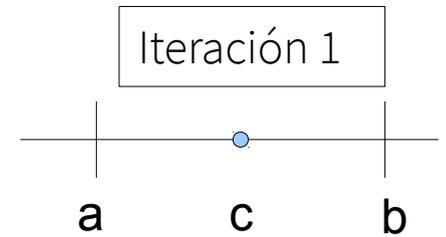
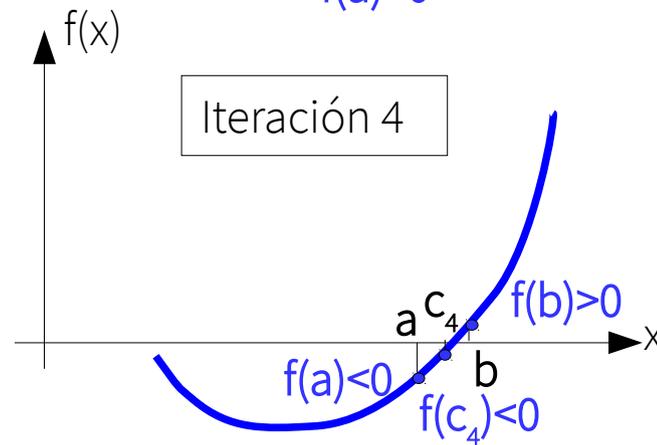
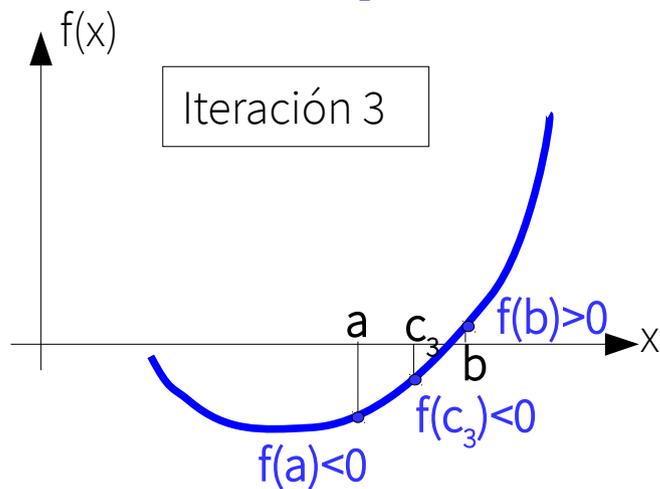
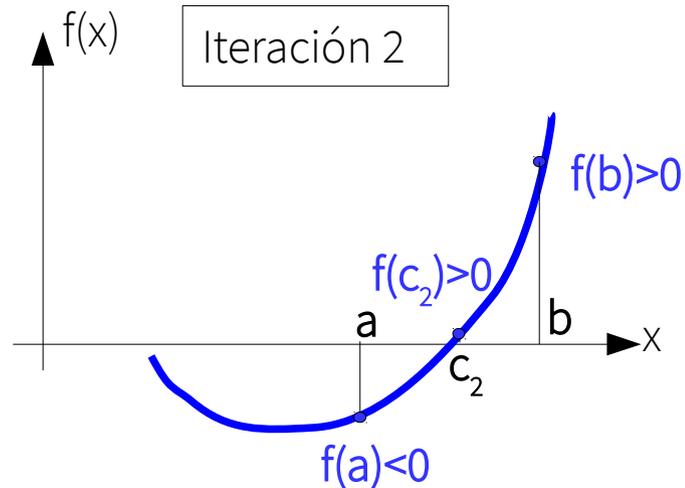
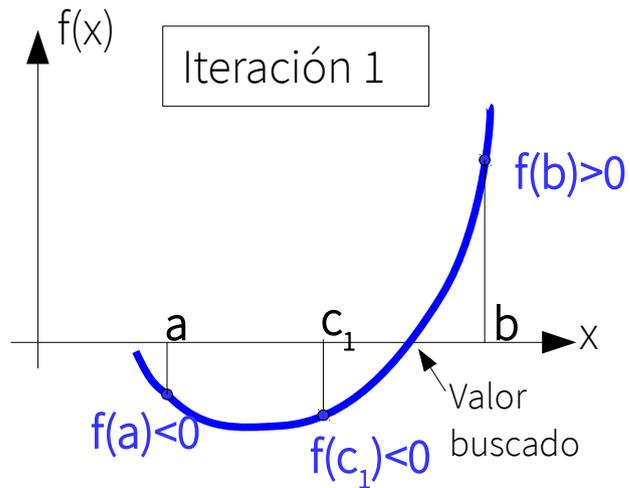
Para la bisección, el **proceso** consiste en **tomar el punto medio** del intervalo, es decir, $c=(a+b)/2$ y luego se analizan las siguientes **posibilidades**:

- $f(a)$ y $f(c)$ tienen signos opuestos, entonces el cero se ubica dentro de $[a,c]$
- $f(c)$ y $f(b)$ tienen signos opuestos, entonces el cero se ubica dentro de $[c,b]$

Al elegir el nuevo intervalo se redujo el intervalo original por la mitad, y se redujo el error a la mitad.

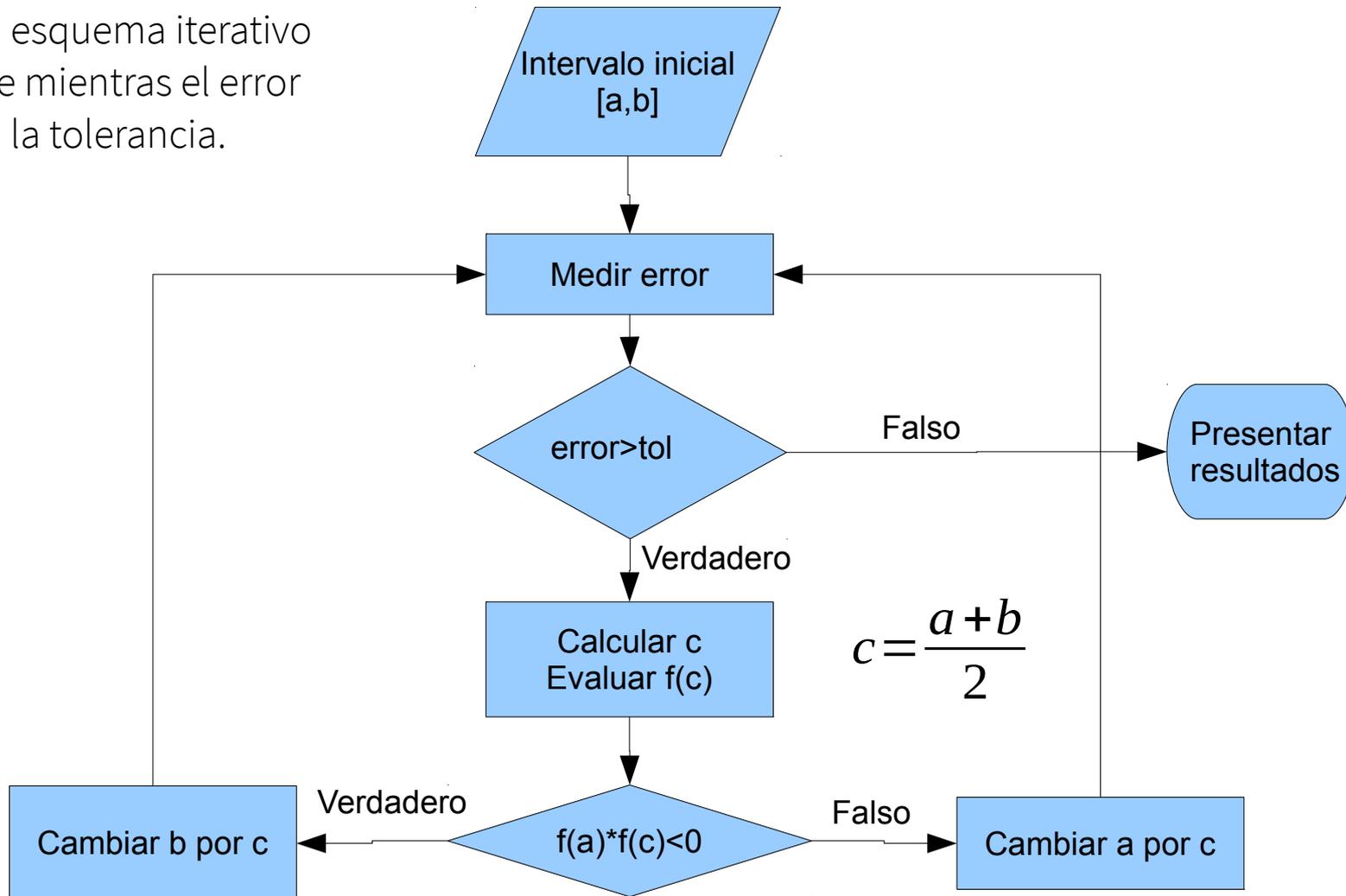
Para continuar el proceso, **se renombra el intervalo $[a,c]$ o $[c,b]$ como $[a,b]$ y se repite la subdivisión.**

Algoritmo de la bisección



Algoritmo de la bisección

Se utiliza un esquema iterativo que se repite mientras el error sea mayor a la tolerancia.



Ejemplo

Ejemplo: $f(x)=x^2+x-1 = 0$ con una tolerancia de 0,001

$$c = \frac{(a+b)}{2} \quad \text{error} = |f(c)|$$

iteración	a	b	f(a)	f(b)	c	f(c)
1	0	1	-1	1	0,5	-0,25
2	0,5	1	-0,25	1	0,75	0,3125

Como $f(a)*f(c)>0$, la raíz está en $[c,b]$. Hay que actualizar el valor de a por el de c.

¿Y ahora? ¿dónde está la raíz? ¿En $[a,c]$ o $[c,b]$?

Ejemplo

Ejemplo: $f(x)=x^2+x-1 = 0$ con una tolerancia de 0,001

$$c = \frac{(a+b)}{2} \quad \text{error} = |f(c)|$$

iteración	a	b	f(a)	f(b)	c	f(c)
1	0	1	-1	1	0,5	-0,25
2	0,5	1	-0,25	1	0,75	0,3125
3	0,5	0,75	-0,25	0,3125	0,625	0,0156
4	0,5	0,625	-0,25	0,0156	0,5625	-0,1211
5	0,5625	0,625	-0,1211	0,0156	0,5938	-0,05371
6	0,5938	0,625	-0,0537	0,0156	0,6094	-0,0193
7	0,6094	0,625	-0,0193	0,0156	0,6172	-0,0019
8	0,6172	0,625	-0,0019	0,0156	0,6211	0,0069
9	0,6172	0,6211	-0,0019	0,0069	0,6192	0,0026
10	0,6172	0,6192	-0,0019	0,0026	0,6182	0,0004

Se alcanzó la tolerancia en 10 iteraciones.

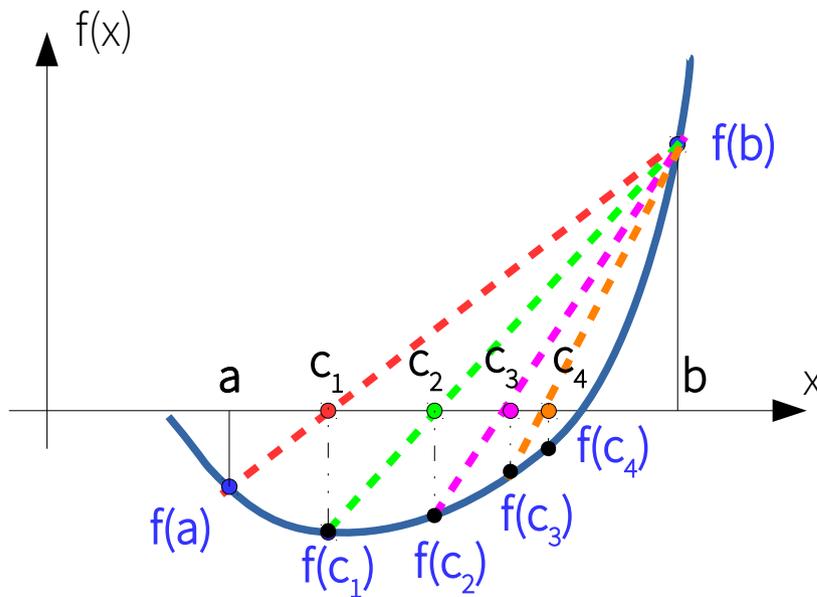
Solución analítica $x_1 = -0,5 + \sqrt{1,25} = 0,61803$

Algoritmo de Regula Falsi

Consiste en una mejora de la velocidad de convergencia del proceso de bisección.

En vez de dividir el intervalo a la mitad, se introduce otro criterio de selección de la nueva aproximación. Se hace **pasar una recta entre a y b** y se determina el **punto de la recta cuya imagen es cero**. Dicho punto será la nueva aproximación de la raíz.

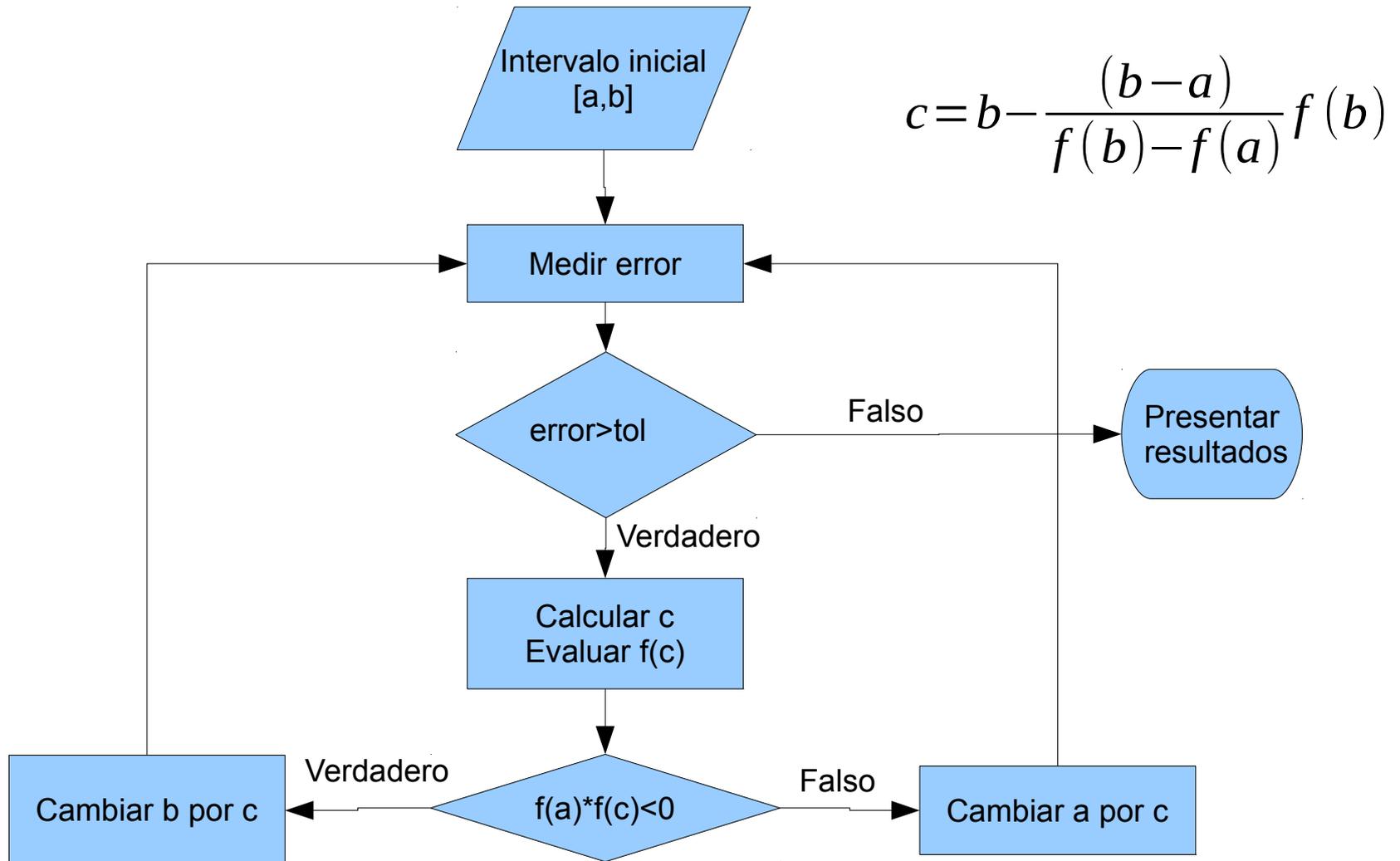
El nombre “Regula Falsi” proviene del latín y significa “posición falsa” debido a que c corresponde a la raíz de la recta, en vez de la función $f(x)$.



$$c = b - \frac{1}{m} f(b)$$

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

Algoritmo de Regula Falsi



Ejemplo

Ejemplo: $f(x)=x^2+x-1 = 0$ con una tolerancia de 0,001

$$c = b - \frac{(b-a)}{f(b)-f(a)} f(b) \quad \text{error} = |f(c)|$$

iteración	a	b	f(a)	f(b)	c	f(c)
1	0	1	-1	1	0,5	-0,25
2	0,5	1	-0,25	1	0,6	-0,04
3	0,6	1	-0,04	1	0,6154	-0,0059
4	0,6154	1	-0,0059	1	0,6176	-0,0009

← Se alcanzó la tolerancia en 4 iteraciones.

Solución analítica $x_1 = -0,5 + \sqrt{1,25} = 0,61803$

Algoritmo de Newton-Raphson

Derivación de la función de iteración de Newton-Raphson

La linealización de la función $f(x)$ alrededor de un punto x_i es:

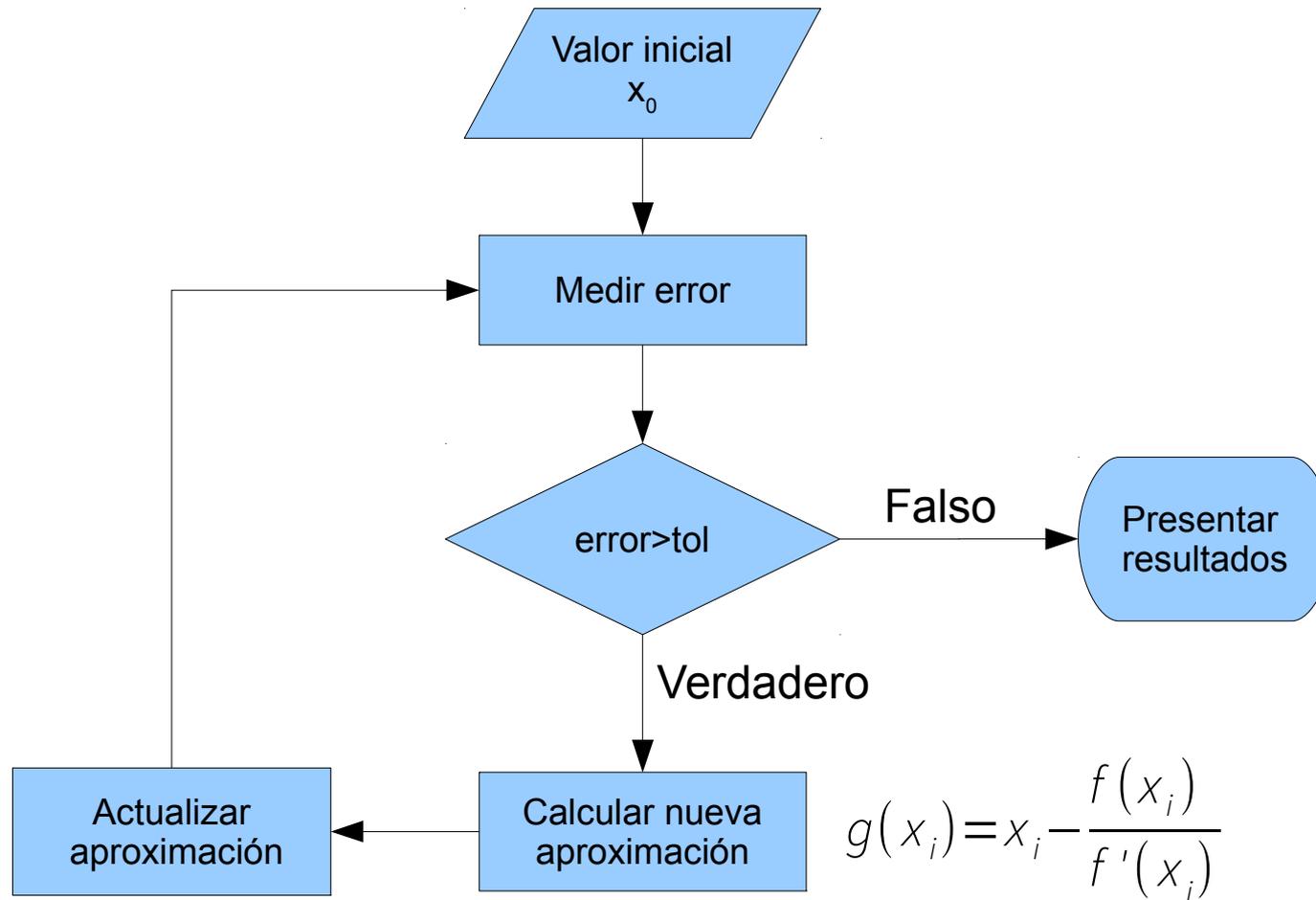
$$L(x) = f(x_i) + f'(x_i)(x - x_i)$$

Si consideramos que la imagen de la linealización es cero para un cierto punto x tenemos que:

$$0 = f(x_i) + f'(x_i)(x_{NR} - x_i) \rightarrow \text{podemos despejar } x_{NR}(x_i) = x_i - \frac{f(x_i)}{f'(x_i)}$$

La función $g(x_i) = x_i - \frac{f(x_i)}{f'(x_i)}$ se llama **función de iteración de Newton-Raphson**

Método de Newton-Raphson



Método de Newton-Raphson

Ejemplo: $f(x)=x^2+x-1 = 0$ con una tolerancia de 0,001

iteración	x_i	$f(x_i)$	$f'(x_i)$	x_{i+1}
1	0	-1	1	1
2	1	1	3	0,6667
3	0,6667	0,1111	2,333	0,6190
4	0,6190	0,0023	2,2381	0,61803
5	0,61803	1,0265E-06	2,2361	---



Se alcanzó la tolerancia en 4 iteraciones. Pero el error es mucho menor!!!

Método de Newton-Raphson

Ejemplo: $f(x)=e^{-x}=0$ con una tolerancia de 0,001

iteración	x_i	f	f'	x_{i+1}
1	1	0,3679	-0,3679	2
2	2	0,1353	-0,1353	3
3	3	0,0498	-0,0498	4
4	4	0,0183	-0,0183	5
5	5	0,0067	-0,0067	6
100	1,00E+02	3,72E-44	-3,72E-44	101
500	5,00E+02	7,12E-218	-7,12E-218	501
745	7,45E+02	1E-323	-1E-323	746
746	746	0	0	#DIV/0!

Explota por aritmética de punto flotante.

Además la convergencia es lenta debido a que f y f' son muy parecidas

Método de Newton-Raphson

Ejemplo: $f(x)=\tan(x)=0$ con una tolerancia de 0,001

iteración	x_i	f	f'	x_{i+1}
1	1,4500	0,9670	0,3223	-1,5503
2	-1,5503	-0,9979	0,2938	1,8459
3	1,8459	1,0743	0,2269	-2,8891
4	-2,8891	-1,2376	0,1070	8,6784
5	8,6784	1,4561	0,0131	-102,4426
6	-102,4426	-1,5610	0,0001	16281,3694
7	16281,3694	1,5707	0,0000	-416358823,9128
8	-416358823,9128	-1,5708	0,0000	272304878428811000,0000

Oscila alrededor de la solución y luego explota

Método de Newton-Raphson

Ejemplo: $f(x)=\tan(x)=0$ con una tolerancia de 0,001

iteración	x_i	f	f'	x_{i+1}
1	0,5000	0,4636	0,8000	-0,0796
2	-0,0796	-0,0794	0,9937	0,0003
3	0,0003	0,0003	1,0000	0,0000
4	0,0000	0,0000	1,0000	0,0000

Si el punto inicial es próximo a la raíz, converge

Conclusión, el método de Newton-Raphson es más rápido pero es más inestable, dependiendo del valor inicial que se utilice y de las propiedades de la función.

Método de la secante

Un inconveniente adicional del método de Newton-Raphson es la evaluación de la derivada. A veces la función es muy compleja para su evaluación. Una solución es aproximar la derivada de la función con una “diferencia finita hacia atrás”.

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

La fórmula de iteración resultante es:

$$x_{i+1} = x_i - \frac{f(x_i)}{m}, \text{ siendo } m = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

Para iniciar el método se requiere proponer dos puntos x_0 y x_1 .

Método de la secante

Ejemplo: $f(x)=x^2+x-1 = 0$ con una tolerancia de 0,001

iteración	x_i	f	x_{i+1}
	0,0000	-1,0000	
1	1,0000	1,0000	0,5000
2	0,5000	-0,2500	0,6000
3	0,6000	-0,0400	0,6190
4	0,6190	0,0023	0,6180
5	0,6180	0,0000	0,6180