



UNIVERSIDAD
DE GRANADA

Métodos Numéricos II
Grado en Matemáticas
2020/2021



TEMA 1

Resolución numérica de ecuaciones y sistemas no lineales.

Versión 16/3/2021

Índice

1. Introducción

¿Sabemos resolver ecuaciones? ¿Qué se entiende por *resolver* una ecuación? En las matemáticas de las etapas primaria y secundaria nos han enseñado a resolver ecuaciones o, al menos, eso creíamos.

Desde hace mucho sabemos resolver ecuaciones como $3x = 6$ o $5x - 9 = 0$ o $2x + 3 = 0$ que son lineales. También sabemos resolver $3x^2 - 4x + 5 = 0$ que puede tener, según el caso, dos soluciones reales, o una real doble, o dos complejas conjugadas, por ser una ecuación de segundo grado. La fórmula $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ ha quedado grabada en nuestras mentes de forma indeleble.

Otras ecuaciones polinómicas también se pueden resolver. La de tercer grado $ax^3 + bx^2 + cx + d = 0$ fue parcialmente resuelta por el italiano Scipione del Ferro a principios del siglo XVI, pero mantuvo su logro en secreto hasta poco antes de su muerte (1526) en que se lo reveló a su discípulo Antonio María Del Fiore. La historia de la fórmula de resolución de la ecuación cúbica es ciertamente curiosa¹ e intervienen en ella personajes como Tartaglia, Cardano, Ferrari y, más tarde, Viète y Descartes. La resolución de la ecuación de cuarto grado no se hizo esperar. De hecho lo lograron los propios Cardano y Ferrari.



Scipione del Ferro
(1465-1526)



Niccolo Tartaglia
(1499-1557)



Girolamo Cardano
(1501-1576)



Ludovico Ferrari
(1522-1565)



François Viète
(1540-1603)



René Descartes
(1596-1650)

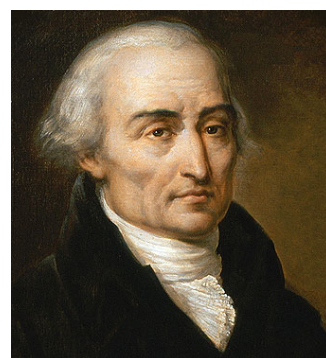
¹véase https://es.wikipedia.org/wiki/Ecuación_de_tercer_grado

A partir del siglo XVI los matemáticos asumieron como reto la búsqueda de fórmulas para resolver ecuaciones de quinto grado o superior. Grandes pesos pesados como Euler, Lagrange y Gauss tuvieron que reconocer su fracaso, aunque sus esfuerzos contribuyeron enormemente a profundizar en el conocimiento de muchas otras ramas de las matemáticas. En 1799 Gauss no hizo sino reforzar el desafío, demostrando en su tesis doctoral (con 22 años) el Teorema Fundamental del Álgebra en el que afirma que todas las ecuaciones de grado n tienen exactamente n soluciones reales o complejas. Pero la fórmula para encontrarlas seguía resistiéndose. No fue sino hasta 1831, casi trescientos años después, en que el jovencísimo matemático francés Évariste Galois, la noche antes de morir en duelo (con 20 años) por un lío de faldas, dejó escritos unos apuntes que han dejado profunda huella en el mundo matemático. Las circunstancias de su vida y su muerte son rocambolescas². Uno de los resultados de sus trabajos, como si fuera un corolario menor, establece que las ecuaciones polinómicas de grado 5 y superiores no son solubles por radicales, lo que en otras palabras significa que no existe la tan pretendida fórmula basada en operaciones algebraicas como suma, resta, multiplicación, división y extracción de raíces.

Si la cuestión de la resolución de ecuaciones polinómicas fue zanjada de la peor forma, más aún en el espacio mucho más abierto de las ecuaciones no polinómicas. En suma, si por *resolver* se entiende *despejar la incógnita* aislándola en un miembro de la igualdad mientras que en el otro miembro quede una expresión computable, esto no va a ser posible en general (ejemplo $xe^x = 5$), como no lo es para ecuaciones polinómicas a partir del grado 5. En consecuencia, hay que ir más en la dirección de resolver



Leonhard Euler
(1707-1783)



Joseph L. de Lagrange
(1736-1813)



Carl Friedrich Gauss
(1777-1855)



Évariste Galois
(1811-1832)

²https://es.wikipedia.org/wiki/Évariste_Galois

aproximadamente, es decir, de encontrar una aproximación numérica a la solución pretendida, buscando el equilibrio entre máxima precisión numérica y mínimo esfuerzo de computación. Eso es lo que entenderemos por *resolución numérica* de ecuaciones.

Planteamiento general.

En general, el problema es hallar las soluciones reales (s) de una ecuación de la forma:

$$f(x) = 0, \quad f : \Omega \subseteq \mathbb{R} \rightarrow \mathbb{R}.$$

Cualquier solución s también se denomina **cero** o **raíz** de la función $f(x)$. El objetivo es la construcción de una sucesión $x_0, x_1, \dots, x_n, \dots$ de aproximaciones tales que $\lim_{n \rightarrow \infty} x_n = s$, donde s es solución real de la ecuación. El procedimiento para obtener cada término de la sucesión viene dirigido por un algoritmo o *método numérico*. Por lo general cada término se calcula a partir del anterior o de los anteriores, por lo que el primer término de la sucesión x_0 debe ser suministrado por algún medio externo al método. En tal caso x_0 recibe el nombre de **aproximación inicial** o también **semilla**.

Conviene distinguir entre raíces simples y múltiples.

Definición 1 (*multiplicidad de raíces*)

- Se dice que s es una raíz o cero de $f(x) = 0$ con **multiplicidad** $m \geq 1$ ($m \in \mathbb{N}$) si existe una función $q(x)$ continua en un entorno de s tal que $f(x) = (x - s)^m q(x)$ y $q(s) \neq 0$.
- Para funciones suficientemente derivables, una raíz real s de $f(x) = 0$ se dice que tiene multiplicidad $m \geq 1$ si $f'(s) = f''(s) = \dots = f^{(m-1)}(s) = 0$ y $f^{(m)}(s) \neq 0$.
- Una raíz con $m = 1$ se llama **simple**, con $m = 2$ **doble**, y en general con $m \geq 2$ **múltiple**.

1.1. Velocidad de convergencia

Antes de proceder a describir métodos de resolución numérica damos algunas definiciones relacionadas con la rapidez de convergencia de sucesiones. Más concretamente,

Definición 2 (*Orden de convergencia*)

1. Diremos que una sucesión positiva $\{\varepsilon_n\}_{n \geq 0}$ convergente hacia cero tiene orden de convergencia $p \geq 1$ si existe una constante $C > 0$ tal que

$$\lim_{n \rightarrow \infty} \frac{\varepsilon_{n+1}}{\varepsilon_n^p} = C \quad (\text{con } C < 1 \text{ en el caso } p = 1).$$

2. Sea $\{x_n\}_{n \geq 0}$ una sucesión real que converge al valor s .
 - i. Diremos que converge con orden al menos p si existe una sucesión $\{\varepsilon_n\}_{n \geq 0}$ convergente a cero con orden p y tal que $|x_n - s| \leq \varepsilon_n$, $n \geq n_0$.
 - ii. Diremos que el orden de convergencia es exactamente p si la sucesión de errores absolutos $\{|x_n - s|\}_{n \geq 0}$ converge hacia cero con orden p ; es decir,

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - s|}{|x_n - s|^p} = C \quad (\text{con } C < 1 \text{ en el caso } p = 1).$$

- La constante C se conoce como constante asintótica del error.
- El caso $p = 1$ se conoce como orden lineal o convergencia lineal.
- El caso $p = 2$ se conoce como orden cuadrático o convergencia cuadrática.

Interpretación práctica del orden de convergencia.

- Si la convergencia es lineal, al avanzar un término en la sucesión se gana a largo plazo un número fijo de cifras de precisión, que depende de la constante asintótica del error.

Ejemplo. La sucesión $\{x_n = 1 + 10^{-n}\}$ tiende linealmente a 1 con $C = 0.1$. En cada término se gana un dígito con respecto al anterior.

$$\begin{aligned}x_0 &= 2 \\x_1 &= 1.1 \\x_2 &= 1.01 \\x_3 &= 1.001 \\x_4 &= 1.0001 \\&\vdots\end{aligned}$$

Ejemplo. La sucesión $\{x_n = 1 + 10^{-2n}\}$ tiende linealmente a 1 con $C = 0.01$. En cada término se ganan dos dígitos con respecto al anterior (compruébese).

$$\begin{aligned}x_0 &= 2 \\x_1 &= 1.01 \\x_2 &= 1.0001 \\x_3 &= 1.000001 \\x_4 &= 1.00000001 \\&\vdots\end{aligned}$$

Ejemplo. La sucesión $\{x_n = 1 + 10^{-\frac{n}{2}}\}$ tiende linealmente a 1. Cada dos términos se gana un dígito (demuéstrese y encuéntrase el valor de C).

- Si la convergencia es cuadrática, avanzando en la sucesión un término se duplica a largo plazo el número de cifras de precisión.

Ejemplo. La sucesión $\{x_n = 1 + 10^{-2^n}\}$ tiende cuadráticamente a 1 con $C = 1$. En cada término se duplica la cantidad de dígitos de precisión con

respecto al anterior (compruébese).

$$\begin{aligned}
 x_0 &= 1.1 \\
 x_1 &= 1.01 \\
 x_2 &= 1.0001 \\
 x_3 &= 1.00000001 \\
 x_4 &= 1.0000000000000001 \\
 &\vdots
 \end{aligned}$$

- Si la convergencia es cúbica, avanzando en la sucesión un término se triplica a largo plazo el número de cifras de precisión.
- Hay una gran diferencia práctica entre orden lineal y cuadrático, pero no tanta entre el cuadrático y órdenes superiores.

Ejercicio. Complete la siguiente tabla, teniendo en cuenta que si $\lim_{n \rightarrow \infty} \frac{\varepsilon_{n+1}}{\varepsilon_n^p} = C$ entonces a la larga se tiene $\varepsilon_{n+1} \approx C\varepsilon_n^p$.

p	C	a la larga se gana...
1	0.1	un dígito en cada nuevo término
1	0.01	dos dígitos en cada nuevo término
1	10^{-d}	d dígitos en cada nuevo término
1		un dígito cada dos nuevos términos
1		un dígito cada m nuevos términos
1		d dígitos cada m nuevos términos
1	$\frac{1}{2}$	
1	k	
2	1	
3	1	
2	10^{-2}	
2	100	

2. Métodos elementales: bisección.

Este método se basa en una demostración constructiva del teorema de Bolzano.

Teorema 1 (*Bolzano*)

Si $f : [a, b] \rightarrow \mathbb{R}$ es continua y $f(a)f(b) < 0$, entonces existe al menos un cero de $f(x)$ en $]a, b[$; es decir, existe $s \in]a, b[$ tal que $f(s) = 0$.

El método de Bisección se basa en aproximar s por el centro del intervalo $m = \frac{a+b}{2}$. Después divide el intervalo en dos mitades y se repite en la mitad que siga cumpliendo las condiciones de Bolzano. De esta forma construye una sucesión de intervalos encajados

$$[a, b] = [a_0, b_0] \supset [a_1, b_1] \supset \cdots \supset [a_n, b_n] \supset [a_{n+1}, b_{n+1}] \supset \cdots$$

todos cumpliendo las condiciones del teorema de Bolzano, de manera que para cada $n = 0, 1, \dots$

- se toma $m_n = \frac{a_n + b_n}{2}$;
- si $f(m_n) = 0$ entonces $s = m_n$ y el algoritmo termina;
- si $f(a_n)f(m_n) < 0$ entonces el nuevo intervalo es $[a_{n+1}, b_{n+1}] = [a_n, m_n]$;
- si $f(m_n)f(b_n) < 0$ entonces el nuevo intervalo es $[a_{n+1}, b_{n+1}] = [m_n, b_n]$;

El algoritmo prosigue hasta que se cumpla algún criterio de parada, y se toma $s \approx m_n$.

Teorema 2 (*Propiedades del método de Bisección*)

- La sucesión $\{m_n\}_{n \geq 0}$ construida por el método de Bisección, converge hacia un cero $s \in]a, b[$ de $f(x)$;
- La sucesión de errores $\{e_n = m_n - s\}_{n \geq 0}$ cumple $|e_n| < \frac{1}{2^{n+1}}(b - a)$;
- El orden de convergencia de $\{m_n\}_{n \geq 0}$ es, al menos, lineal ($p = 1$).

Demostración.

1. Si $\ell_n = b_n - a_n$ entonces es fácil ver que $\ell_n = \frac{\ell_{n-1}}{2} = \dots = \frac{\ell_0}{2^n} = \frac{b-a}{2^n}$, de donde $\lim_{n \rightarrow \infty} \ell_n = 0$ y, por tanto, $\lim_{n \rightarrow \infty} m_n = \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = s' \in]a, b[$. Además se puede probar que s' es un cero de f .
2. Los errores sucesivos cumplen $|e_n| = |m_n - s| < \frac{\ell_n}{2} \Rightarrow |e_n| < \frac{b-a}{2^{n+1}}$.
3. La sucesión $\{\ell_n\}$ converge a cero linealmente con constante asintótica del error $C = \frac{1}{2}$ y acota a la sucesión $\{e_n\}$, luego ésta última converge a cero al menos linealmente.

□

El Cuadro ?? muestra los resultados de aplicar el método de bisección a la ecuación $x^3 + x - 3 = 0$ en el intervalo de partida $[1, 2]$. Se han realizado iteraciones hasta cumplir con el criterio $|e_n| \leq 10^{-6}$.

Cálculo previo del número de iteraciones necesarias. El método de bisección permite, al contrario que otros métodos, calcular previamente el número de iteraciones necesarias para obtener la solución con error inferior a uno dado. En efecto, si se parte del intervalo $[a, b]$ y se desea la solución con error inferior a ε , entonces podemos imponer

$$|e_n| < \frac{b-a}{2^{n+1}} \leq \varepsilon,$$

de donde

$$2^{n+1} \geq \frac{b-a}{\varepsilon} \Rightarrow n \geq \log_2 \frac{b-a}{\varepsilon} - 1 = \frac{\ln \frac{b-a}{\varepsilon}}{\ln 2} - 1$$

Obsérvese que este cálculo no depende en absoluto de la ecuación particular, sino tan solo del intervalo inicial y el error máximo impuesto.

Ejemplo. Para una precisión de $\varepsilon = 10^{-6}$ (seis cifras decimales) partiendo del intervalo $[1, 2]$ el número de iteraciones que lo garantizan es:

$$n \geq \frac{\ln \frac{1}{10^{-6}}}{\ln 2} - 1 = 18.9316 \dots \Rightarrow \boxed{n=19}.$$

Este cálculo previo del número de iteraciones garantiza una aproximación a la solución dentro del error requerido, pero no se descarta que alguna

METODO DE BISECCION					
n	a	b	m	$f(m)$	err
0	1.00000000	2.00000000	1.50000000	1.88e+00	5.00e-01
1	1.00000000	1.50000000	1.25000000	2.03e-01	2.50e-01
2	1.00000000	1.25000000	1.12500000	-4.51e-01	1.25e-01
3	1.12500000	1.25000000	1.18750000	-1.38e-01	6.25e-02
4	1.18750000	1.25000000	1.21875000	2.90e-02	3.12e-02
5	1.18750000	1.21875000	1.20312500	-5.53e-02	1.56e-02
6	1.20312500	1.21875000	1.21093750	-1.34e-02	7.81e-03
7	1.21093750	1.21875000	1.21484375	7.77e-03	3.91e-03
8	1.21093750	1.21484375	1.21289062	-2.82e-03	1.95e-03
9	1.21289062	1.21484375	1.21386719	2.47e-03	9.77e-04
10	1.21289062	1.21386719	1.21337891	-1.77e-04	4.88e-04
11	1.21337891	1.21386719	1.21362305	1.15e-03	2.44e-04
12	1.21337891	1.21362305	1.21350098	4.84e-04	1.22e-04
13	1.21337891	1.21350098	1.21343994	1.53e-04	6.10e-05
14	1.21337891	1.21343994	1.21340942	-1.21e-05	3.05e-05
15	1.21340942	1.21343994	1.21342468	7.05e-05	1.53e-05
16	1.21340942	1.21342468	1.21341705	2.92e-05	7.63e-06
17	1.21340942	1.21341705	1.21341324	8.54e-06	3.81e-06
18	1.21340942	1.21341324	1.21341133	-1.80e-06	1.91e-06
19	1.21341133	1.21341324	1.21341228	3.37e-06	9.54e-07

Cuadro 1: Método de Bisección aplicado a la ecuación $x^3 + x - 3 = 0$ que tiene una única raíz simple en $[1, 2]$. Cota de error: 10^{-6} .

```
#####
#### script para ilustrar el metodo de biseccion ####
#####
clc;
function y = f(x)
    y = x^3+x-3;
endfunction
a = 1;
b = 2;
eps = 1e-6;
fa = f(a);
fb = f(b);
disp('METODO DE BISECCION');
disp(' n      a          b          m          f(m)      err');
formato = '%3d %6.8f %6.8f %6.8f %9.2e %9.2e\n';
n=0;
do
    m=(a+b)/2; fm = f(m);
    printf(formato,n,a,b,m,fm,(b-a)/2);
    if fa*fm < 0
        b=m; fb=fm;
    else
        a=m; fa=fm;
    endif
    n++;
until b-a <= eps
```

Cuadro 2: Código Octave empleado para el Cuadro ??.

iteración anterior pudiera estar más cerca de s que la última obtenida. Dicho de otro modo, la sucesión de errores e_n no tiene por qué ser decreciente en valor absoluto. Como contraejemplo se puede considerar el intervalo $[0, 1]$ con la raíz $s = 0.5000001$ y una cota de error de $\varepsilon = 10^{-6}$. La garantía la darán 19 iteraciones, pero la primera de ellas ya cumple con el criterio.

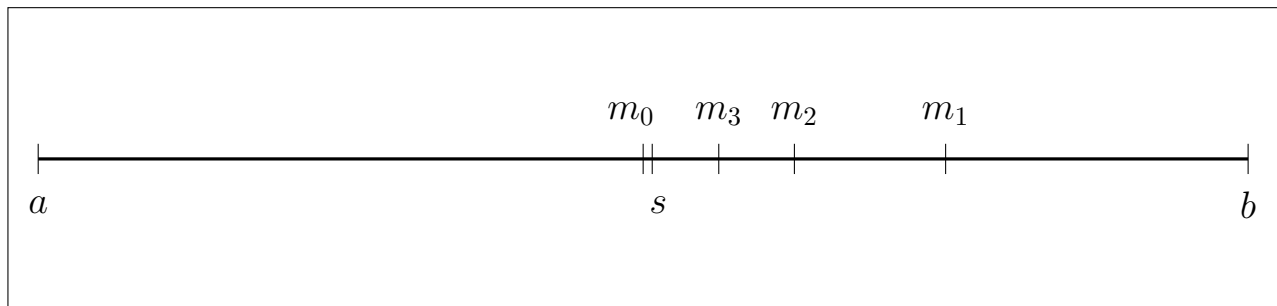
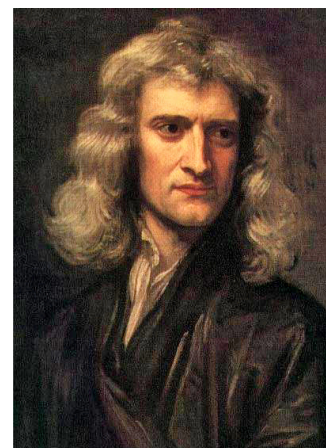


Figura 1: Método de Bisección: la sucesión de errores no tiene que ser decreciente.

3. Métodos de Newton-Raphson y secante. Convergencia.

El orden de convergencia del método de bisección es lineal con constante asintótica del error $C = \frac{1}{2}$, lo que se puede interpretar como una ganancia de un dígito de precisión cada tres iteraciones aproximadamente, lo cual es muy lento. Existen métodos mucho más rápidos.

El método de Newton-Raphson³ o de las tangentes construye una sucesión $\{x_n\}_{n \geq 0}$ de aproximaciones así:



Isaac Newton
(1643-1727)

$$\begin{aligned} x_0 &= \text{aproximación inicial o semilla} \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, \dots \end{aligned}$$

Ejemplo. Para la ecuación $x^2 - 5 = 0$ cuya raíz positiva es $\sqrt{5}$ la iteración es $x_{n+1} = x_n - \frac{x_n^2 - 5}{2x_n} = \frac{1}{2} \left(x_n + \frac{5}{x_n} \right)$. Partiendo de $x_0 = 2$ como semilla tenemos

n	x_n
1	2.2500000000000000
2	2.2361111111111111
3	2.236067977915804
4	2.236067977499790
5	2.236067977499790

donde se puede ver que entre x_4 y x_5 la diferencia está fuera de la vista. Todos los decimales que se muestran en x_4 y x_5 coinciden con los de $\sqrt{5}$. De hecho x_3 ya presenta los 9 primeros decimales exactos.

La justificación deductiva del método de Newton-Raphson se puede hacer de forma analítica o geométrica.

Deducción analítica. Si f es suficientemente derivable con continuidad se

³Joseph Raphson (1648-1715), matemático inglés, publicó el método 46 años antes que Newton.

puede escribir su desarrollo de Taylor en el punto x_n

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \text{Error}$$

si $x \approx x_n$ podemos despreciar el error

$$f(x) \approx f(x_n) + f'(x_n)(x - x_n)$$

y la solución (aproximada) puede obtenerse haciendo $f(x) = 0$, de donde

$$f(x_n) + f'(x_n)(x - x_n) \approx 0$$

y despejando x tendríamos

$$x \approx x_n - \frac{f(x_n)}{f'(x_n)}$$

que tomaríamos como x_{n+1} .

Deducción geométrica. x_{n+1} es la abscisa del punto de intersección del eje X con la recta tangente a la curva $(x, f(x))$ en el punto $(x_n, f(x_n))$ (véase la Figura ??).

Teorema 3 (*Convergencia global de NR*)

Dada la ecuación $f(x) = 0$ con $f \in \mathcal{C}^2([a, b])$ verificando

1. $f(a)f(b) < 0$
2. $f'(x) \neq 0 \forall x \in [a, b]$
3. $f''(x)$ no cambia de signo en $[a, b]$
4. $\max \left\{ \left| \frac{f(a)}{f'(a)} \right|, \left| \frac{f(b)}{f'(b)} \right| \right\} \leq b - a$

Entonces

1. La ecuación admite una única raíz real $s \in]a, b[$
2. El método de Newton-Raphson converge a s para toda aproximación inicial $x_0 \in [a, b]$
3. El orden de convergencia del método es, al menos, 2.

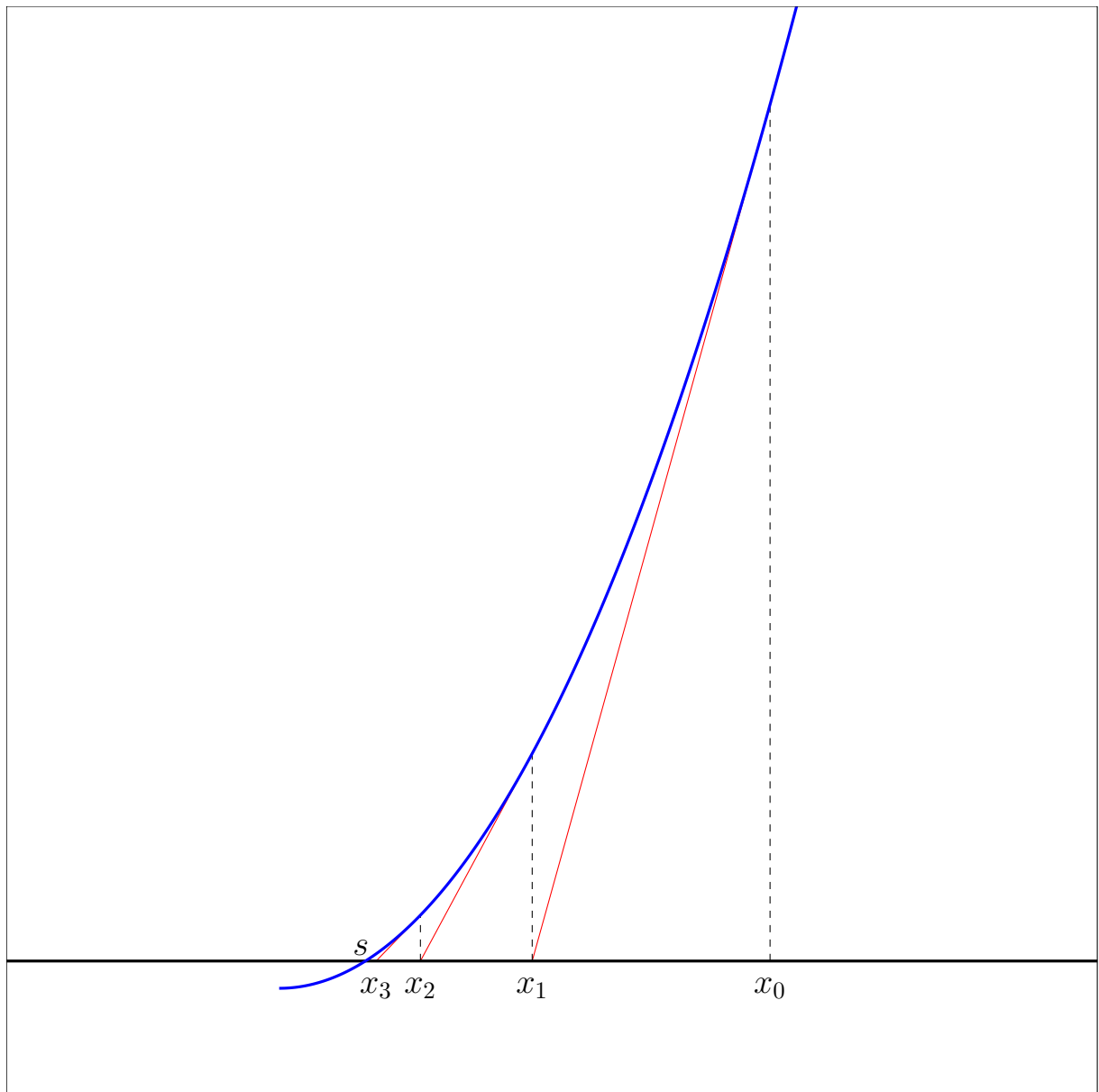


Figura 2: Ilustración gráfica del método de Newton Raphson.

Demostración.

- Por (1) y Bolzano, la función admite al menos un cero $s \in]a, b[$. Por (2), f es estrictamente monótona y no podrá anularse una segunda vez, luego s es único.
- Sea $g(x) = x - \frac{f(x)}{f'(x)}$ la función asociada al método. Entonces $g'(x) = \frac{f(x)f''(x)}{(f'(x))^2}$.
- Supongamos $f' > 0$, $f'' \geq 0$. Entonces g decrece en $[a, s]$ y crece en $[s, b]$ (aunque no lo haga estrictamente).
- Si $a \leq x_0 < s$ entonces $s = g(s) \leq x_1 = g(x_0) \leq g(a) = a - \frac{f(a)}{f'(a)} \leq b$ por (4), luego $x_1 \in [s, b]$.
- Si $s < x_0 \leq b$ entonces $s = g(s) \leq x_1 = g(x_0) \leq g(b) = b - \frac{f(b)}{f'(b)} < b$, luego $x_1 \in [s, b]$. En consecuencia, toda la sucesión $\{x_n\}$ estará contenida en $[s, b]$ salvo eventualmente la semilla x_0 .
- Si $x_0 = s$ entonces $x_1 = s$.
- Por otro lado en $]s, b]$ ocurre que $x_{n+1} = g(x_n) < x_n$. Tenemos, por tanto, una sucesión monótona decreciente (salvo la semilla) y acotada inferiormente, que ha de tener un límite s' que verifica $s' = g(s')$, luego ha de ser $s' = s$.
- Otras combinaciones de signos se estudian y demuestran de forma análoga.
- El orden de convergencia se deduce por Taylor.

$$e_{n+1} = x_{n+1} - s = x_n - s - \frac{f(x_n)}{f'(x_n)} = e_n + \frac{f(s) - f(x_n)}{f'(x_n)} \stackrel{\text{Taylor orden 2}}{=} \frac{f''(\xi_n)e_n^2}{2f'(x_n)}$$

con ξ_n comprendido entre s y x_n . Por tanto $\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^2} = \frac{1}{2} \left| \frac{f''(s)}{f'(s)} \right| > 0$ (salvo que $f''(s) = 0$). En consecuencia, la convergencia será, al menos, cuadrática con constante asintótica del error $C = \frac{1}{2} \left| \frac{f''(s)}{f'(s)} \right|$. □

Teorema 4 (*Relajación de condiciones del t. de conv. global*)

En ausencia de la condición 4 del Teorema ??, éste sigue siendo válido si $f(x_0)f''(x_0) > 0$.

Teorema 5 (*Convergencia local*)

Dada la ecuación $f(x) = 0$ con $f \in \mathcal{C}^2(I)$ siendo I un entorno abierto de una solución s simple ($f'(s) \neq 0$), entonces existe un subintervalo centrado $I_\varepsilon =]s - \varepsilon, s + \varepsilon[\subseteq I$ para el cual la sucesión generada por el método de Newton-Raphson converge a s con orden al menos cuadrático $\forall x_0 \in I_\varepsilon$.

Los criterios de parada usuales, dada una tolerancia $T > 0$, son:

- absoluto: $|x_{n+1} - x_n| < T$, que depende de la magnitud de los términos, y
- relativo: $\frac{|x_{n+1} - x_n|}{|x_{n+1}|} < T$, que no depende de la magnitud de los términos.

METODO DE NEWTON-RAPHSON			
n	x_n	$ x_{n+1} - x_n $	$\frac{ x_{n+1} - x_n }{ x_{n+1} }$
0	1.50000000		
1	1.25806452	2.42e-01	1.92e-01
2	1.21470533	4.34e-02	3.57e-02
3	1.21341279	1.29e-03	1.07e-03
4	1.21341166	1.12e-06	9.26e-07
5	1.21341166	8.48e-13	6.99e-13

Cuadro 3: Método de Newton-Raphson aplicado a la ecuación $x^3 + x - 3 = 0$ con $x_0 = 1.5$ y tolerancia $T = 10^{-6}$.

3.1. Comportamiento del Método de Newton-Raphson para raíces múltiples.

Cuando la raíz buscada s no es simple, el método de Newton-Raphson sigue convergiendo, pero pierde el orden cuadrático de convergencia. El Cuadro ?? muestra un ejemplo.

```
#####
### script para ilustrar el metodo de Newton-Raphson ###
#####
clc;
function y = f(x)
    y = x^3+x-3;
endfunction
function y = f1(x) % la derivada
    y = 3*x^2+1;
endfunction
x0 = 1.5;
tol = 1e-6;
disp('METODO DE NEWTON-RAPHSON');
disp(' n      x_n      dif.abs    dif.rel');
printf('%3d %6.8f\n',0,x0);
n = 0;
x1 = x0;
do
    n++;
    x0 = x1;
    x1=x0-f(x0)/f1(x0);
    printf('%3d %6.8f %9.2e %9.2e\n',n,x1,abs(x1-x0),abs(x1-x0)/abs(x1));
until abs(x1-x0) <= tol
```

Cuadro 4: Código Octave empleado para el Cuadro ??.

Teorema 6 (*Convergencia local para raíces múltiples*)

Dada la ecuación $f(x) = 0$ con $f \in \mathcal{C}^2(I)$ siendo I un entorno abierto de una solución s de multiplicidad $m > 1$, entonces existe un subintervalo centrado $I_\varepsilon =]s - \varepsilon, s + \varepsilon[\subseteq I$ para el cual la sucesión generada por el método de Newton-Raphson converge a s con orden lineal ($p = 1$) y con constante asintótica del error $C = 1 - \frac{1}{m}$.

Demostración (Sugerencia). Se considera $f(x) = (x - s)^m q(x)$ con $q(s) \neq 0$. Sea $g(x) = x - \frac{f(x)}{f'(x)}$ la función asociada al método. Entonces

$$\begin{aligned} g'(x) &= \frac{f(x)f''(x)}{(f'(x))^2} = \dots \\ &= \frac{m(m-1)q^2(x) + 2m(x-s)q(x)q'(x) + (x-s)^2q(x)q''(x)}{(mq(x) + (x-s)q'(x))^2} \end{aligned}$$

y por tanto $g'(s) = 1 - \frac{1}{m}$. □

METODO DE NEWTON-RAPHSON			
n	x_n	$ x_{n+1} - x_n $	$\frac{ x_{n+1} - x_n }{ x_{n+1} }$
0	0.50000000		
1	0.72925296	2.29e-01	3.14e-01
2	0.85852527	1.29e-01	1.51e-01
3	0.92759526	6.91e-02	7.45e-02
4	0.96336080	3.58e-02	3.71e-02
5	0.98156853	1.82e-02	1.85e-02
6	0.99075596	9.19e-03	9.27e-03
7	0.99537086	4.61e-03	4.64e-03
8	0.99768364	2.31e-03	2.32e-03
9	0.99884137	1.16e-03	1.16e-03
10	0.99942058	5.79e-04	5.80e-04
11	0.99971026	2.90e-04	2.90e-04
12	0.99985512	1.45e-04	1.45e-04
13	0.99992756	7.24e-05	7.24e-05
14	0.99996378	3.62e-05	3.62e-05
15	0.99998189	1.81e-05	1.81e-05
16	0.99999094	9.06e-06	9.06e-06
17	0.99999547	4.53e-06	4.53e-06
18	0.99999774	2.26e-06	2.26e-06
19	0.99999887	1.13e-06	1.13e-06
20	0.99999943	5.66e-07	5.66e-07

Cuadro 5: Método de Newton-Raphson aplicado a la ecuación $e^{1-x} + x - 2 = 0$ que tiene una raíz doble en $x = 1$, con $x_0 = 0.5$ y tolerancia 10^{-6} .

```
#####
### metodo de Newton-Raphson con raices multiples ###
#####
clc;
function y = f(x)
    y = exp(1-x)+x-2;
endfunction
function y = f1(x) % la derivada
    y = -exp(1-x)+1;
endfunction
x0 = 0.5;
tol = 1e-6;
disp('METODO DE NEWTON-RAPHSON');
disp(' n      x_n      dif.abs    dif.rel');
printf('%3d %6.8f\n',0,x0);
n = 0;
x1 = x0;
do
    n++;
    x0 = x1;
    x1=x0-f(x0)/f1(x0);
    printf('%3d %6.8f %9.2e %9.2e\n',n,x1,abs(x1-x0),abs(x1-x0)/abs(x1));
until abs(x1-x0) <= tol
```

Cuadro 6: Código Octave empleado para el Cuadro ??.

Recuperación de la convergencia cuadrática. Se puede conseguir de dos maneras diferentes.

1. Usando el método $x_{n+1} = g_m(x_n)$ generado por $g_m(x) = x - m \frac{f(x)}{f'(x)}$. Para hacerlo es necesario conocer previamente la multiplicidad m , por lo cual el interés de esta opción es relativo.
2. Aplicando Newton-Raphson a la ecuación $\mu(x) = 0$ con $\mu(x) = \frac{f(x)}{f'(x)}$, lo que conduce al método $x_{n+1} = h(x_n)$ generado por

$$h(x) = x - \frac{\mu(x)}{\mu'(x)} = x - \frac{f(x)f'(x)}{(f'(x))^2 - f(x)f''(x)}.$$

En este segundo caso hace falta que f sea al menos de clase 2. Se puede demostrar que si s es raíz de f entonces también lo es de μ pero simple.

Ambos métodos modificados se comparan en el Cuadro ??.

METODO DE NEWTON-RAPHSON				
n	$x_n(1)$	$ x_{n+1} - x_n $	$x_n(2)$	$ x_{n+1} - x_n $
0	0.50000000		0.50000000	
1	0.95850592	4.59e-01	1.04929971	5.49e-01
2	0.99971305	4.12e-02	1.00039848	4.89e-02
3	0.99999999	2.87e-04	1.00000003	3.98e-04
4	0.99999999	0.00e+00	0.99999998	4.59e-08

Cuadro 7: Dos métodos corregidos de Newton-Raphson para raíces múltiples.

```
#####
#### metodo de Newton-Raphson con raices multiples ####
#####
clc;
function y = f(x)
    y = exp(1-x)+x-2;
endfunction
function y = f1(x) % la derivada
    y = -exp(1-x)+1;
endfunction
function y = f2(x) % la derivada
    y = exp(1-x);
endfunction
x0 = 0.5;
z0 = 0.5;
tol = 1e-6;
disp('METODO DE NEWTON-RAPHSON');
disp(' n      x_n      dif.abs      x_n      dif.abs');
printf('%3d %6.8f      %6.8f\n',0,x0,z0);
n = 0;
x1 = x0;
z1 = z0;
do
    n++;
    x0 = x1;
    z0 = z1;
    x1 = x0 - 2*f(x0)/f1(x0);
    z1 = z0 - f(z0)*f1(z0)/(f1(z0)^2-f(z0)*f2(z0));
    printf('%3d %6.8f %9.2e %6.8f %9.2e\n',n,x1,abs(x1-x0),z1,abs(z1-z0));
until (abs(x1-x0) <= tol) && (abs(z1-z0) <= tol)
```

Cuadro 8: Código Octave empleado para el Cuadro ??.

3.2. El método de la secante.

Se trata de una alternativa interesante al método de Newton-Raphson cuando no se puede disponer de la derivada o bien no interesa. Se construyen las sucesivas aproximaciones como

$$\begin{aligned} x_0, x_1 &= \text{aproximaciones iniciales o semillas} \\ x_{n+1} &= x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} = \frac{x_{n-1}f(x_n) - x_nf(x_{n-1})}{f(x_n) - f(x_{n-1})}, \quad n \geq 1. \end{aligned}$$

En el Cuadro ?? se muestra un ejemplo.

METODO DE LA SECANTE		
n	x_n	$ x_{n+1} - x_n $
0	1.00000000	
1	2.00000000	1.00e+00
2	1.12500000	8.75e-01
3	1.17798165	5.30e-02
4	1.21562415	3.76e-02
5	1.21335829	2.27e-03
6	1.21341158	5.33e-05
7	1.21341166	7.93e-08

Cuadro 9: Método de la Secante aplicado a la ecuación $x^3 + x - 3 = 0$ con $x_0 = 1, x_1 = 2$ y tolerancia 10^{-6} .

Dedución analítica. Se puede obtener a partir del método de Newton Raphson sustituyendo $f'(x_n)$ por su aproximación como cociente diferencial $f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$. Compruébese.

Dedución geométrica. x_{n+1} es la abscisa del punto de intersección del eje X con la recta secante a la curva $(x, f(x))$ en los puntos $(x_{n-1}, f(x_{n-1}))$ y $(x_n, f(x_n))$. Compruébese.

Observaciones

- No se tienen en cuenta cambios de signo.

```
#####
#### metodo de la secante ####
#####
clc;
function y = f(x)
    y = x^3+x-3;
endfunction
x0 = 1; x1 = 2; f0 = f(x0);
tol = 1e-6;
disp('METODO DE LA SECANTE');
disp(' n      x_n      dif.abs');
printf('%3d %6.8f\n',0,x0);
n = 0;
x2 = x1; x1 = x0; f1 = f0;
do
    n++;
    x0 = x1; f0 = f1;
    x1 = x2; f1 = f(x1);
    x2=x1-f1*(x1-x0)/(f1-f0);
    printf('%3d %6.8f %9.2e\n',n,x1,abs(x1-x0));
until abs(x1-x0) <= tol
```

Cuadro 10: Código Octave empleado para el Cuadro ??. Se ha optimizado para realizar una sola evaluación de la función en cada iteración.

- Puede ocurrir que $f(x_n) - f(x_{n-1})$ sea cero o muy próximo, lo cual es problemático porque puede provocar fuerte cancelación de dígitos iguales, y en caso extremo la cancelación total, generando una división por cero.

Teorema 7 (*Convergencia local para el método de la secante*)

Dada la ecuación $f(x) = 0$ con $f \in \mathcal{C}^2(I)$ siendo I un entorno abierto de una solución s simple ($f'(s) \neq 0$), entonces existe un subintervalo centrado $I_\varepsilon =]s - \varepsilon, s + \varepsilon[\subseteq I$ para el cual la sucesión generada por el método de la Secante converge a s con orden al menos $p = \frac{1+\sqrt{5}}{2} \approx 1.62$ (superlineal) $\forall x_0 \in I_\varepsilon$.

Demostración (Sugerencia). De la fórmula de interpolación de Newton de f en los nodos x_n, x_{n-1} con término de error podemos escribir

$$f(x) = f[x_n] + f[x_n, x_{n-1}](x - x_n) + f[x_n, x_{n-1}, x](x - x_n)(x - x_{n-1})$$

y tomando $x = s$, despejando $f(x_n)$ y sustituyendo adecuadamente se obtiene

$$e_{n+1} = e_n e_{n-1} \frac{f[x_{n-1}, x_n, s]}{f[x_{n-1}, x_n]} = e_n e_{n-1} \frac{f''(\xi_{2,n})}{2f'(\xi_{1,n})}$$

de donde se puede deducir el resultado. □

4. Métodos de iteración funcional. Convergencia.

Muchos métodos para resolver $f(x) = 0$ consisten en transformar el problema en uno equivalente de la forma $x = g(x)$ y de ahí construir el método $x_{n+1} = g(x_n)$ con una semilla x_0 adecuada. Un valor s verificando $s = g(s)$ se denomina *punto fijo* de g . Por ejemplo, la ecuación $x^3 + 9x + 9 = 0$ que tiene una única raíz real en el intervalo $[-1, 0]$ se puede reescribir de muy diversas formas:

- $x = g_1(x) = -\frac{x^3}{9} - 1,$
- $x = g_2(x) = -\frac{9}{x^2 + 9},$
- $x = g_3(x) = -\sqrt{-\frac{9}{x} - 9},$
- $x = g_4(x) = \frac{2x^3 - 9}{3(x^2 + 3)}.$

n	$x = g_1(x)$	$x = g_2(x)$	$x = g_3(x)$	$x = g_4(x)$
0	-1	-1	-1	-1
1	-0.888889	-0.9	0	-0.916667
2	-0.921963	-0.917431	\nexists	-0.914909
3	-0.912924	-0.914478		-0.914908
4	-0.915460	-0.914981		-0.914908
5	-0.914754	-0.914895		-0.914908

Definición 3 (*Función contráctil*)

Una función $g : [a, b] \rightarrow \mathbb{R}$ cumpliendo $|g(x) - g(y)| \leq L|x - y| \quad \forall x, y \in [a, b]$ se dirá **lipschitziana**. Esta condición se denomina **condición de Lipschitz** y L es la **constante de Lipschitz**. En el caso $L < 1$ la función g se dirá **contráctil**, **contractiva** o **contracción** y L la **constante de contractibilidad**.

Teorema 8 (del Punto Fijo)

Sea $g : [a, b] \rightarrow [a, b]$ contráctil con constante de contractibilidad L . Entonces

1. Existe un único punto fijo s de g en $[a, b]$.
2. El método $x_{n+1} = g(x_n)$ genera una sucesión convergente a $s \forall x_0 \in [a, b]$
3. Los errores $e_n = x_n - s$ cumplen $|e_n| \leq \frac{L^n}{1-L} |x_1 - x_0| \quad n, 1, 2, \dots$

Demostración.

1. La existencia de s se prueba aplicando Bolzano a $f(x) = g(x) - x$. La unicidad se deduce de la condición de contractibilidad.
2. Es fácil probar que $\forall x_0 \in [a, b]$ la sucesión $\{x_n\}_{n \geq 0} \subset [a, b]$ converge a s :

$$|x_n - s| \leq \dots \leq L^n |x_0 - s|.$$

3. Como $|x_0 - s| \leq |x_1 - x_0| + |x_1 - s| \leq |x_1 - x_0| + L|x_0 - s|$, entonces

$$|x_0 - s| \leq \frac{1}{1-L} |x_1 - x_0| \quad \text{y por tanto} \quad |x_n - s| \leq \frac{L^n}{1-L} |x_1 - x_0|.$$

□

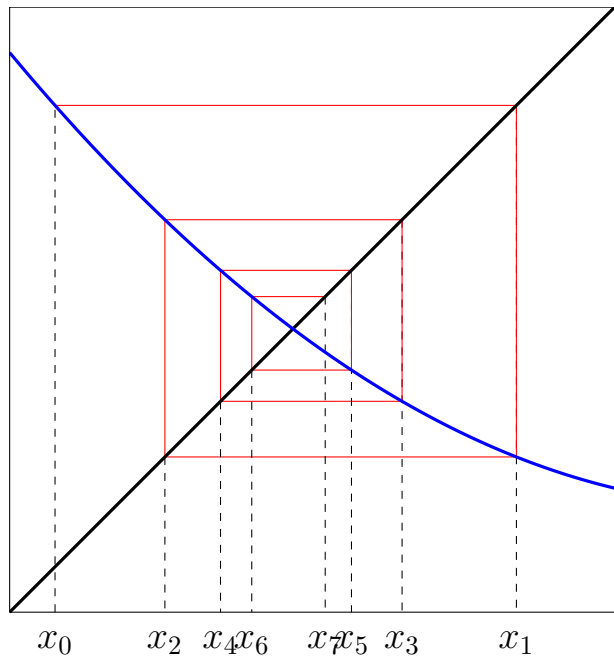
Observaciones.

- Para funciones derivables, el teorema sigue siendo válido si se sustituye la contractibilidad por $|g'(x)| \leq L < 1 \quad \forall x \in [a, b]$.
- Si $g : \mathbb{R} \rightarrow \mathbb{R}$ cumple la condición anterior, el teorema se sigue cumpliendo.

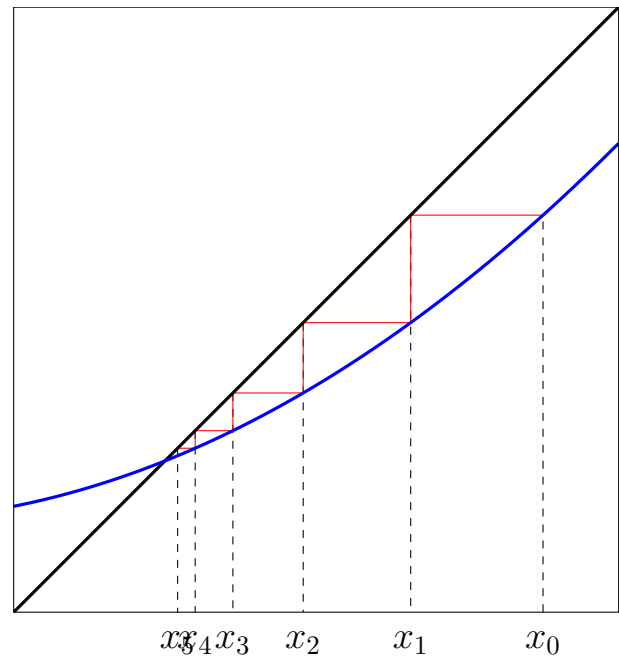
Teorema 9 (Convergencia local)

Sea $g \in \mathcal{C}^1(I)$ siendo I un entorno abierto de un punto fijo s de g tal que $|g'(s)| < 1$. Entonces existe un subintervalo $I_\varepsilon \subseteq I$ en el que el método $x_{n+1} = g(x_n)$ converge a $s \quad \forall x_0 \in I_\varepsilon$.

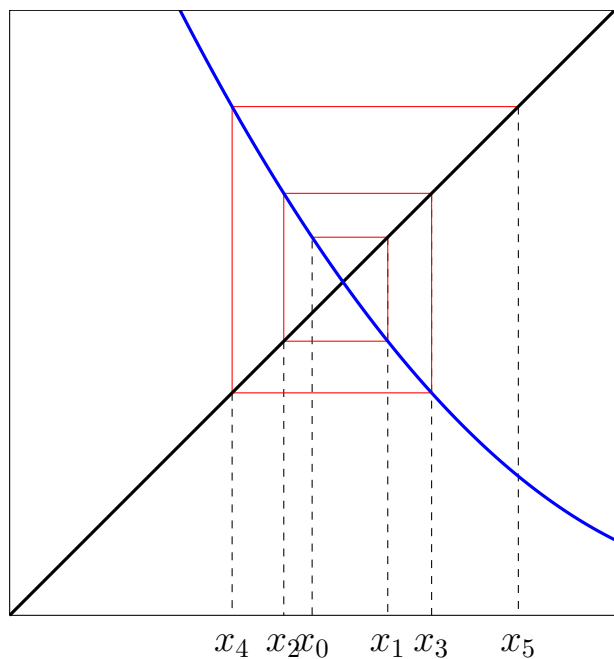
Interpretación geométrica. Véase la Figura ??.



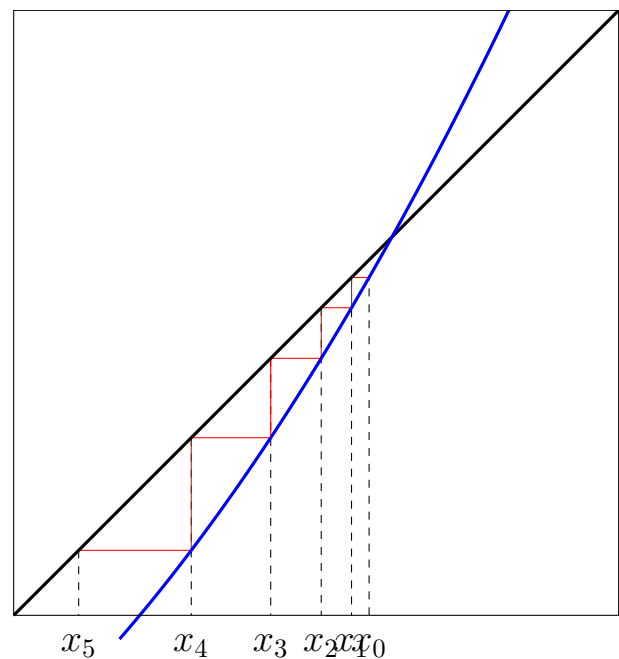
a. Convergencia en espiral u oscilante



b. Convergencia en escalera o monótona



c. Divergencia en espiral u oscilante



d. Divergencia en escalera o monótona

Figura 3: Ejemplos de convergencia y divergencia en iteración de punto fijo

METODO DE ITERACION FUNCIONAL			
n	g_1	g_2	g_3
0	1.000000	1.000000	1.000000
1	0.281718	0.693147	0.806824
2	1.674595	0.835884	0.792135
3	-2.336633	0.772012	0.792060
4	2.903347	0.801099	0.792060
5	-15.235084	0.787958	0.792060
6	3.000000	0.793916	0.792060
7	-17.085532	0.791219	0.792060
8	3.000000	0.792441	0.792060
9	-17.085536	0.791887	0.792060

Cuadro 11: Métodos de iteración funcional aplicados a la ecuación

$$e^x + x - 3 = 0 \text{ con } x_0 = 1, g_1(x) = 3 - e^x, g_2(x) = \ln(3 - x), \text{ y}$$

$$g_3(x) = \frac{xe^x - e^x + 3}{e^x + 1}.$$

Error y orden de convergencia. En la práctica no es fácil obtener la constante de contractibilidad L . Si g es suficientemente derivable podríamos escribir $x_{n+1} - s = g(x_n) - g(s) = g'(\xi_n)(x_n - s)$ con ξ_n comprendido entre x_n y s de manera que $e_{n+1} = g'(\xi_n)e_n$, y por tanto $\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|} = |g'(s)|$.

Si $g'(s) \neq 0$ se tendrá convergencia local lineal con constante $C = |g'(s)|$. De lo contrario, si $g \in \mathcal{C}^2(I)$ y $g''(s) \neq 0$, un razonamiento similar conduce a

$$e_{n+1} = \frac{1}{2}g''(\xi_n)e_n^2, \text{ y por tanto } \lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^2} = \frac{1}{2}|g''(s)|$$

con lo que la convergencia sería localmente cuadrática con constante asintótica del error $C = \frac{1}{2}|g''(s)|$. De forma más general tenemos el siguiente

Teorema 10 (*Orden de convergencia*)

Sea $s = g(s)$ con $g \in \mathcal{C}^p(I)$ verificando $g'(s) = \dots = g^{(p-1)}(s)$ y $g^{(p)}(s) \neq 0$. Entonces para x_0 suficientemente próximo a s el método iterativo $x_{n+1} = g(x_n)$ converge a s con orden de convergencia p . Además, la constante asintótica del error es $C = \frac{1}{p!}|g^{(p)}(s)|$.


```
#####
#### comparacion de metodos de iteracion funcional ####
#####
clc;
function y = g1(x)
    y = 3-exp(x);
endfunction
function y = g2(x)
    y = log(3-x);
endfunction
function y = g3(x)
    y = (x*exp(x)-exp(x)+3)/(exp(x)+1);
endfunction
x0 = 1;
y0 = 1;
z0 = 1;
disp('METODOS DE ITERACION FUNCIONAL');
disp(' n      g_1      g_2      g_3');
formato = '%2d %10.6f %10.6f %10.6f\n';
printf(formato,0,x0,y0,z0);
for n = 1:9
    x0 = g1(x0);
    y0 = g2(y0);
    z0 = g3(z0);
    printf(formato,n,x0,y0,z0);
endfor
```

Cuadro 12: Código Octave empleado para el Cuadro ??.

4.1. Aceleración Δ^2 de Aitken

Se trata de un método que permite acelerar la convergencia de una sucesión que converge linealmente. A partir de la sucesión $\{x_n\}_{n \geq 0}$ se construye la sucesión acelerada $\{\hat{x}_n\}_{n \geq 0}$ como

$$\hat{x}_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n} = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n} = \frac{x_{n+2}x_n - x_{n+1}^2}{x_{n+2} + x_n - 2x_{n+1}}$$

siendo $\Delta x_n = x_{n+1} - x_n$ el operador *diferencia progresiva*, supuesto lineal y por tanto $\Delta^2 x_n = \Delta(\Delta x_n) = x_{n+2} - 2x_{n+1} + x_n$.

En algunos textos (Atkinson, Ralston, Wikipedia,...) se presenta la formulación del método Δ^2 de Aitken como

$$\hat{x}_{n+2} = x_{n+2} - \frac{(\Delta x_{n+1})^2}{\Delta^2 x_n} = x_{n+2} - \frac{(x_{n+2} - x_{n+1})^2}{x_{n+2} - 2x_{n+1} + x_n} = \frac{x_{n+2}x_n - x_{n+1}^2}{x_{n+2} + x_n - 2x_{n+1}}$$

que es equivalente, salvo por la numeración de subíndices de la sucesión acelerada. Aquí usaremos la primera (Gasca, Burden).

Teorema 11 (*Aceleración de Aitken*)

Sea la sucesión $\{x_n\}_{n \geq 0}$ convergente a s al menos linealmente. Entonces la sucesión acelerada $\hat{x}_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$ converge a s más rápidamente en el sentido $\lim_{n \rightarrow \infty} \frac{\hat{e}_n}{e_n} = 0$ siendo $\hat{e}_n = \hat{x}_n - s$ y $e_n = x_n - s$.

Demostración. Supongamos $\frac{e_{n+1}}{e_n} = \lambda_n$ con $\lim_{n \rightarrow \infty} |\lambda_n| = C < 1$. Como $\Delta e_n = \Delta x_n$, se tiene

$$\begin{aligned} \hat{e}_n &= e_n - \frac{(\Delta e_n)^2}{\Delta^2 e_n} = e_n - \frac{e_n^2(\lambda_n - 1)^2}{e_n(\lambda_{n+1}\lambda_n - 2\lambda_n + 1)} \\ &= e_n \frac{\lambda_{n+1}\lambda_n - 2\lambda_n + 1 - \lambda_n^2 + 2\lambda_n - 1}{\lambda_{n+1}\lambda_n - 2\lambda_n + 1} \\ &= e_n \frac{\lambda_n \Delta \lambda_n}{\lambda_{n+1}\lambda_n - 2\lambda_n + 1}, \text{ luego } \lim_{n \rightarrow \infty} \frac{\hat{e}_n}{e_n} = 0. \end{aligned}$$

□

4.2. Aceleración de Steffensen

Cuando la sucesión $\{x_n\}_{n \geq 0}$ convergente a s proviene de un método de iteración funcional $x_{n+1} = g(x_n)$, entonces se puede combinar éste con el procedimiento de aceleración de Aitken para aumentar la velocidad de convergencia. El procedimiento de aceleración de Steffensen que parte de x_0 consiste en usar g hasta tener tres términos, acelerar mediante Aitken y volver a generar tres términos mediante g para volver a acelerar. Más concretamente:

$$\left. \begin{array}{l} x_0 \\ x_1 = g(x_0) \\ x_2 = g(x_1) \end{array} \right\} \longrightarrow x'_0 = x_0 - \frac{(\Delta x_0)^2}{\Delta^2 x_0}$$

$$\left. \begin{array}{l} x'_0 \\ x'_1 = g(x'_0) \\ x'_2 = g(x'_1) \end{array} \right\} \longrightarrow x''_0 = x'_0 - \frac{(\Delta x'_0)^2}{\Delta^2 x'_0}$$

y así sucesivamente. La sucesión $\{x_0^{(n)}\}$ convergerá a s aún más rápidamente que la de Aitken.

ACELERACIONES DE AITKEN Y STEFFENSEN		
g	Aitken	Steffensen
0.5000000000		
0.6065306597		
0.5452392119	0.5676238764	0.5676238764
0.5797030949	0.5672989893	
0.5600646279	0.5671931424	
0.5711721490	0.5671593645	0.5671433141
0.5648629470	0.5671484533	
0.5684380476	0.5671449524	
0.5664094527	0.5671438247	0.5671432904
0.5675596343	0.5671434623	
0.5669072129	0.5671433457	
0.5672771960	0.5671433082	0.5671432904

Cuadro 13: Aceleraciones de Aitken y Steffensen aplicadas a la función $g(x) = e^{-x}$ con $x_0 = 0.5$.

5. Ecuaciones polinómicas.

Se aborda aquí el problema de resolver $P(x) = 0$ siendo $P(x)$ un polinomio en x . Evidentemente son válidos todos los métodos de aplicación general para ecuaciones (Bisección, Newton-Raphson, Secante, etc.). El tratamiento específico de una ecuación polinómica pasa por tres etapas:

1. Determinación del número de raíces reales
2. Localización y separación en intervalos disjuntos de las raíces reales
3. Estimación precisa de cada raíz mediante un método numérico adecuado

Las dos primeras etapas se abordan con la teoría de Sturm (véase Gasca, Cálculo Numérico, UNED, 1996). Para la tercera es muy adecuado adaptar el método de Newton-Raphson.

```
#####
#### aceleracion de Aitken y de Steffensen ####
#####
clc;
function y = g(x)
    y = exp(-x);
endfunction
x0 = 0.5; x1 = g(x0); x2 = g(x1);
a0 = x0 - (x1-x0)^2/(x2-2*x1+x0);
s0 = a0;
disp('ACELERACIONES DE AITKEN Y STEFFENSEN');
disp('      g      Aitken      Steffensen');
formato = '%14.10f';
printf([formato '\n'],x0);
printf([formato '\n'],x1);
printf([formato formato formato '\n'],x2,a0,s0);
for n = 1:3
    for j = 1:2
        x0 = x1; x1 = x2; x2 = g(x1);
        a0 = x0 - (x1-x0)^2/(x2-2*x1+x0);
        printf([formato formato '\n'],x2,a0);
    endfor
    x0 = x1; x1 = x2; x2 = g(x1);
    a0 = x0 - (x1-x0)^2/(x2-2*x1+x0);
    s1 = g(s0); s2 = g(s1);
    s0 = s0 - (s1-s0)^2/(s2-2*s1+s0);
    printf([formato formato formato '\n'],x2,a0,s0);
endfor
```

Cuadro 14: Código Octave empleado para el Cuadro ??.

Teorema 12 (acotación de raíces)

Sea $P(x) = a_k x^k + a_{k-1} x^{k-1} + \dots + a_1 x + a_0$. Toda raíz s de P verifica $|s| \leq 1 + \alpha$ donde $\alpha = \max_{0 \leq i < k} \left| \frac{a_i}{a_k} \right|$.

Demostración. $P(s) = 0 \Rightarrow s^k = -\sum_{i=0}^{k-1} \frac{a_i}{a_k} s^i \Rightarrow |s|^k \leq \alpha \sum_{i=0}^{k-1} |s|^i = \alpha \frac{|s|^k - 1}{|s| - 1}$. Si fuese $|s| > \alpha + 1$ sería $\alpha < |s| - 1$ y entonces $|s|^k < |s|^k - 1$, absurdo. \square

Definición 4 (Sucesión de Sturm)

$\{f_0, f_1, \dots, f_m\} \subset \mathcal{C}[a, b]$ es una sucesión de Sturm en $[a, b]$ si verifica $\forall x \in [a, b]$:

1. $f_0 \in \mathcal{C}^1[a, b]$,
2. $f_0(x) = 0 \Rightarrow f'_0(x) f_1(x) > 0$,
3. $f_j(x) = 0 \Rightarrow f_{j-1}(x) f_{j+1}(x) < 0 \quad j = 1, \dots, m-1$.
4. $f_m(x) \neq 0$.

Teorema 13 (Sturm)

Si $\{f_0, f_1, \dots, f_m\}$ es una sucesión de Sturm en $[a, b]$, entonces la cantidad de raíces reales que f_0 tiene en $[a, b]$ es la diferencia entre el número de cambios de signo que hay en la secuencia $\{f_0(a), f_1(a), \dots, f_m(a)\}$ y el que hay en la secuencia $\{f_0(b), f_1(b), \dots, f_m(b)\}$

Demostración. (Véase la ilustración en la Figura ??). Supongamos que x recorre el intervalo desde a hasta b . Al atravesar x un cero de f_j , $j = 1, \dots, m-1$ no se altera el número de cambios de signo en la secuencia $\{f_0(x), f_1(x), \dots, f_m(x)\}$. Por otro lado, al atravesar x un cero de f_0 hay dos casos posibles: 1, pasar de $-$ a $+$ con lo que $f_1 > 0$, o 2, pasar de $+$ a $-$ con lo que $f_1 < 0$. En cualquier caso se pierde un cambio de signo. \square

Procedimiento de obtención de una sucesión de Sturm para un polinomio. Sea $P(x)$ un polinomio.

1. $f_0 = P$, $f_1 = P'$.

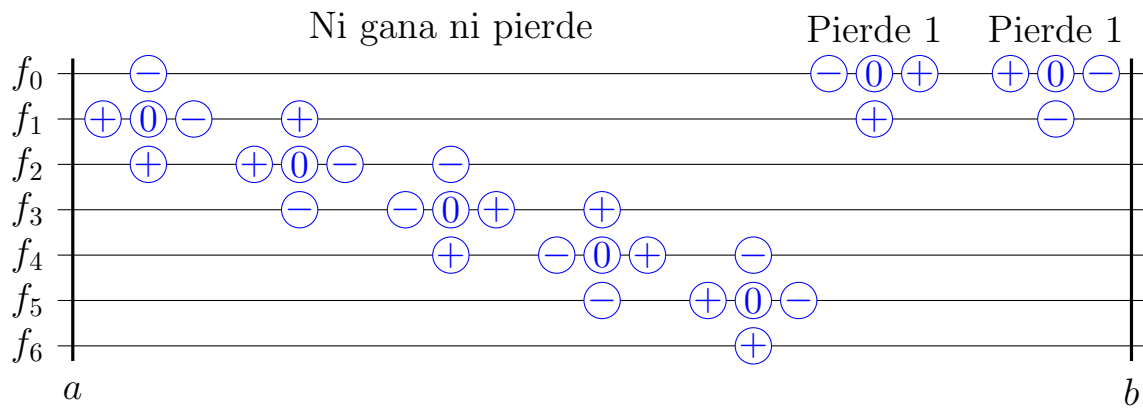


Figura 4: Ilustración de la demostración del Teorema de Sturm.

2. $f_{j-2} = q_{j-1}f_{j-1} - f_j$, $j = 2, \dots, m$ (algoritmo de Euclides de división, con el resto cambiado de signo).
3. f_m = último resto no nulo.
4. Si f_m es constante, entonces $\{f_0, \dots, f_m\}$ es una sucesión de Sturm $\forall [a, b]$; de lo contrario lo es $\left\{ \frac{f_0}{f_m}, \frac{f_1}{f_m}, \dots, \frac{f_m}{f_m} \right\}$.

Ejemplo. Sea $P(x) = 2x^5 - x^4 - 4x^3 + 2x^2 - 6x + 3$.

$$\begin{aligned}
 f_0 &= P = 2x^5 - x^4 - 4x^3 + 2x^2 - 6x + 3 \\
 f_1 &= P' = 10x^4 - 4x^3 - 12x^2 + 4x - 6 \\
 f_2 &= \dots = 21x^3 - 12x^2 + 59x - 36 \\
 f_3 &= \dots = 115x^2 - 48x + 9 \\
 f_4 &= \dots = -185171x + 118188 \\
 f_5 &= \dots = -9 - \frac{118188 \cdot 4703412}{185171^2} = \text{cte.}
 \end{aligned}$$

Localización y separación. $\alpha = \max\{\frac{1}{2}, \frac{4}{2}, \frac{2}{2}, \frac{6}{2}, \frac{3}{2}\} = 3 \Rightarrow |z| \leq 4$.

$$\begin{aligned}
 -4 &\rightarrow \{- + - + - -\} \text{ 4 cambios} \\
 4 &\rightarrow \{+ + + + - -\} \text{ 1 cambio} \\
 0 &\rightarrow \{+ - - + + -\} \text{ 3 cambios} \\
 2 &\rightarrow \{+ + + + - -\} \text{ 1 cambio} \\
 1 &\rightarrow \{- - + + - -\} \text{ 2 cambios}
 \end{aligned}$$

Por tanto hay 3 raíces reales localizadas en $[-4, 0]$, $[0, 1]$ y $[1, 2]$

5.1. El método de Newton-Raphson para ecuaciones polinómicas.

Se puede mejorar la eficiencia del método mediante algoritmos de evaluación para P y P' . Para evaluar $P(x) = a_k x^k + a_{k-1} x^{k-1} + \dots + a_1 x + a_0$ en el punto $x = t$ se puede escribir $P(x) = (x - t)Q(x) + b_0$ donde $Q(x) = b_k x^{k-1} + \dots + b_2 x + b_1$ es el cociente de dividir $P(x)$ entre el binomio $(x - t)$ y b_0 es una constante que, obviamente, es el valor de $P(t)$. El método para obtener estos coeficientes, que no es más que la regla de Ruffini, se conoce como *algoritmo de Horner*:

$$\begin{aligned} b_k &= a_k \\ b_j &= a_j + t \cdot b_{j+1}, \quad j = k-1, \dots, 0 \end{aligned}$$

La misma idea se puede aprovechar para el cálculo eficiente de $P'(t)$. Podemos escribir $P'(x) = Q(x) + (x - t)Q'(x)$. y por tanto $P'(t) = Q(t)$ con lo que para obtener $P'(t)$ basta con aplicar el algoritmo de Horner a $Q(x)$

$$\begin{aligned} c_k &= b_k \\ c_j &= b_j + t \cdot c_{j+1}, \quad j = k-1, \dots, 1. \end{aligned}$$

METODO DE NEWTON-RAPHSON PARA POLINOMIOS		
n	x_n	$ x_{n+1} - x_n $
0	1.500000000	
1	1.25806452	2.42e-01
2	1.21470533	4.34e-02
3	1.21341279	1.29e-03
4	1.21341166	1.12e-06
5	1.21341166	8.48e-13

Cuadro 15: Método de Newton-Raphson aplicado a la ecuación polinómica $x^3 + x - 3 = 0$ con $x_0 = 1.5$ y tolerancia 10^{-6} .

```
#####
#### metodo de Newton-Raphson para polinomios ####
#####
clc;
pol = [1 0 1 -3]; %grado decreciente
x0 = 1.5;
tol = 1e-6;
disp('METODO DE NEWTON-RAPHSON PARA POLINOMIOS');
disp(' n      x_n      dif.abs');
printf('%3d %6.8f\n',0,x0);
k = length(pol);
n = 0;
x1 = x0;
do
    n++;
    x0 = x1;
    b = pol(1);
    c = b;
    for j = 2:k-1
        b = pol(j) + x0*b;
        c = b + x0*c;
    endfor
    b = pol(k) + x0*b;
    x1 = x0 - b/c;
    printf('%3d %6.8f %9.2e\n',n,x1,abs(x1-x0));
until abs(x1-x0) <= tol
```

Cuadro 16: Código Octave empleado para el Cuadro ??.

6. Métodos de resolución para sistemas de ecuaciones no lineales.

Consideraremos aquí el problema de resolver el sistema de ecuaciones $F(X) = 0$ donde $F : D \subseteq \mathbb{R}^k \rightarrow \mathbb{R}^k$ es una función vectorial de las variables reales $X = (x_1, \dots, x_k)$ y componentes $f_1, \dots, f_k : D \subseteq \mathbb{R}^k \rightarrow \mathbb{R}$. Para mayor facilidad supondremos $D = \prod_{i=1}^k [a_i, b_i]$, que es un dominio compacto y convexo de \mathbb{R}^k .

De forma desarrollada el sistema se escribiría

$$\left. \begin{array}{c} f_1(x_1, \dots, x_k) = 0 \\ \vdots \\ f_k(x_1, \dots, x_k) = 0 \end{array} \right\}.$$

6.1. Métodos de iteración funcional.

Siguiendo la idea de la iteración funcional de punto fijo, transformaremos el sistema original en otro equivalente en la forma $X = G(X)$, a partir del cual se construirá el método iterativo $X_{n+1} = G(X_n)$ que parte de la semilla X_0 . De forma desarrollada, el sistema transformado sería

$$\left. \begin{array}{c} x_1 = g_1(x_1, \dots, x_k) \\ \vdots \\ x_k = g_k(x_1, \dots, x_k) \end{array} \right\}.$$

Ejemplo. Consideremos el sistema

$$\left. \begin{array}{l} x^2 + xy - 10 = 0 \\ 3xy^2 + y - 57 = 0 \end{array} \right\} \text{ en } D = [1, 3] \times [2, 4],$$

que tiene una solución en $s = (2, 3) \in D$. Una posible transformación podría ser

$$\left. \begin{array}{l} x = g_1(x, y) = \frac{10}{x+y} \\ y = g_2(x, y) = 57 - 3xy^2 \end{array} \right\}.$$

Partiendo de $(x_0, y_0) = (2.1, 2.9) \approx s$ se obtienen los resultados de la tabla

n	x_n	y_n
0	2.1	2.9
1	2.00	4.02
2	1.66	-39.82
3	-0.26	-7847.86

por lo que está claro que el método no es adecuado.

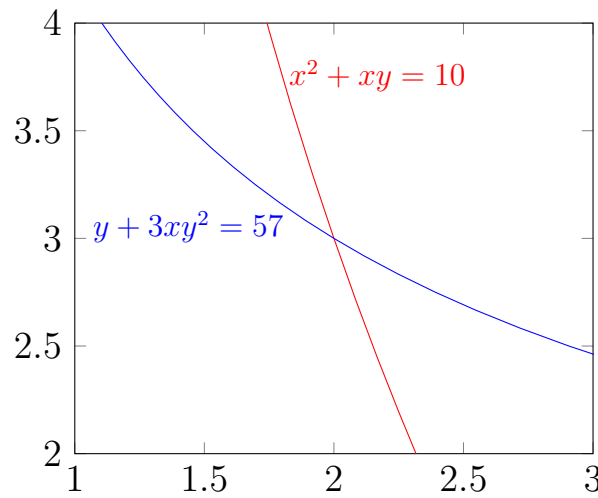


Figura 5: Curvas asociadas a cada una de las ecuaciones del sistema. La solución es el punto de corte.

Teorema 14 (*Convergencia global*)

Sea $G : D \rightarrow D$ verificando para alguna norma $\|\cdot\|$ en \mathbb{R}^k

$$\|G(X) - G(Y)\| \leq L\|X - Y\| \quad \forall X, Y \in D \quad \text{con } 0 \leq L < 1.$$

Entonces,

1. Existe un único punto fijo $S = (s_1, \dots, s_k) \in D$ de G .
2. El método $X_{n+1} = G(X_n)$ converge a S para todo $X_0 \in D$, es decir, $\lim_{n \rightarrow \infty} \|X_n - S\| = 0$.
3. Los errores $E_n = X_n - S$ cumplen $\|E_n\| \leq \frac{L^n}{1 - L} \|X_1 - X_0\| \quad \forall n \geq 1$.

Observaciones.

- Si existen todas las derivadas parciales y son continuas en D , la condición de contractibilidad puede sustituirse por⁴ $\|G'(X)\| \leq L < 1$ o también por $\left| \frac{\partial g_i(X)}{\partial x_j} \right| \leq \frac{L}{k} \forall i, j \forall X \in D$.

Teorema 15 (*Convergencia local*)

Sea I un entorno de $S = G(S)$ y $G \in \mathcal{C}^1(I)$ tal que $\|G'(S)\| < 1$ para alguna norma matricial $\|\cdot\|$. Entonces existe un subentorno $I_\varepsilon \subseteq I$ en el que el método $X_{n+1} = G(X_n)$ converge a S para todo $X_0 \in I_\varepsilon$.

Observaciones.

- Se puede sustituir la condición $\|G'(S)\| < 1$ por $\rho(G'(S)) < 1$ (radio espectral).
- Se puede sustituir la condición $\|G'(S)\| < 1$ por $\left| \frac{\partial g_i(S)}{\partial x_j} \right| \leq \frac{1}{k} \forall i, j$.

6.2. El método de Newton-Raphson para sistemas.

Para resolver el sistema $F(X) = 0$ el método de Newton-Raphson consiste en $X_{n+1} = X_n - J_n^{-1}F(X_n)$, $n = 0, 1, \dots$, donde J_n es la matriz jacobiana

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k}{\partial x_1} & \dots & \frac{\partial f_k}{\partial x_k} \end{pmatrix}$$

evaluada en la aproximación X_n . El método estará bien definido si J_n admite inversa, es decir, si $\det(J_n) \neq 0 \forall n$.

En la práctica no hay que invertir el jacobiano en cada paso, sino resolver

⁴ $G'(X)$ indica la matriz jacobiana:

$$G' = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_1}{\partial x_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_k}{\partial x_1} & \dots & \frac{\partial g_k}{\partial x_k} \end{pmatrix}$$

un sistema de ecuaciones lineales:

$$J_n \Delta X_n = -F(X_n) \longrightarrow X_{n+1} = X_n + \Delta X_n, \text{ siendo } \Delta X_n = X_{n+1} - X_n.$$

Ejemplo. En el caso del Ejemplo anterior, la matriz jacobiana asociada es

$$J = \begin{pmatrix} 2x + y & x \\ 3y^2 & 6xy + 1 \end{pmatrix}$$

que ha de ser invertible en cada aproximación calculada. Esto es cierto en la región D considerada, ya que $\det(J) = 12x^2y + 3xy^2 + 2x + y > 0$ en D . El método de Newton-Raphson quedaría:

$$\begin{aligned} X_0 &= \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 3.5 \end{pmatrix} \text{ (semilla no muy próxima a la solución } s) \\ X_{n+1} &= \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \begin{pmatrix} 2x_n + y_n & x_n \\ 3y_n^2 & 6x_n y_n + 1 \end{pmatrix}^{-1} \begin{pmatrix} x_n^2 + x_n y_n - 10 \\ 3x_n y_n^2 + y_n - 57 \end{pmatrix} \\ & n = 0, 1, 2, \dots \end{aligned}$$

Los resultados numéricos se muestran en el cuadro ??.

METODO DE NEWTON-RAPHSON PARA SISTEMAS			
n	x_n	y_n	$\ x_{n+1} - x_n\ _\infty$
0	1.50000000	3.50000000	
1	2.03602882	2.84387510	6.56e-01
2	1.99870061	3.00228856	1.58e-01
3	1.99999998	2.99999941	2.29e-03
4	2.00000000	3.00000000	5.87e-07

Cuadro 17: Método de Newton-Raphson aplicado al sistema del ejemplo con tolerancia 10^{-6} .

Teorema 16

Sea el sistema de ecuaciones $F(X) = 0$ con $F \in \mathcal{C}^2(I)$ siendo I un entorno de una solución S con $\det(J(F)|_{X=S}) \neq 0$. Entonces existe un entorno $I_\varepsilon \subseteq I$ para el cual la sucesión generada por el método de Newton Raphson converge a la solución con orden de convergencia, al menos, cuadrático $\forall X_0 \in I_\varepsilon$.

```
#####
### metodo de Newton-Raphson para sistemas ###
#####
clc;
function y = f(x)
    y = [x(1)^2+x(1)*x(2)-10; 3*x(1)*x(2)^2+x(2)-57];
endfunction
function y = f1(x) % el jacobiano
    y = [2*x(1)+x(2), x(1); 3*x(2)^2, 6*x(1)*x(2)+1];
endfunction
x0 = [1.5; 3.5];
tol = 1e-6;
disp('METODO DE NEWTON-RAPHSON PARA SISTEMAS');
disp(' n      x_n      dif.abs');
printf('%3d %6.8f %6.8f\n',0,x0);
n = 0;
x1 = x0;
do
    n++;
    x0 = x1;
    x1=x0-f1(x0)\f(x0);
    printf('%3d %6.8f %6.8f %9.2e\n',n,x1,norm(x1-x0,Inf));
until abs(x1-x0) <= tol
```

Cuadro 18: Código Octave empleado para el Cuadro ??.

7. Anexo: sobre normas matriciales.

Dada una norma vectorial $\|\cdot\|_V$ sobre \mathbb{R}^k , se define su *norma matricial subordinada o inducida* como

$$\|A\|_M = \max_{\|x\|_V=1} \|Ax\|_V.$$

La norma subordinada es compatible con la vectorial: $\|Ax\|_V \leq \|A\|_M \|x\|_V$ $\forall A \in \mathbb{R}^{k \times k}, x \in \mathbb{R}^k$. Las normas más populares son

- **Norma 1:** $\|A\|_1 = \max_{j=1,\dots,k} \left\{ \sum_{i=1}^k |a_{ij}| \right\}$. Es la inducida por $\|x\|_1 = \sum_{i=1}^k |x_i|$.
- **Norma ∞ :** $\|A\|_\infty = \max_{i=1,\dots,k} \left\{ \sum_{j=1}^k |a_{ij}| \right\}$. Inducida por $\|x\|_\infty = \max_i |x_i|$.

- **Norma 2:** $\|A\|_2 = \sqrt{\rho(A^T A)}$ donde $\rho(B) = \max_{j=1, \dots, k} |\lambda_j|$ es el radio espectral de B y λ_j es valor propio de B . Es la inducida por la norma euclídea $\|x\|_2 = \sqrt{x_1^2 + \dots + x_k^2}$.

Algunas propiedades que relacionan normas, radio espectral y determinante. Sea A una matriz cuadrada cualquiera y $\|\cdot\|$ una norma matricial cualquiera.

- $\rho(A) \leq \|A\|$.
- $\det(A) = \prod_{j=1}^k \lambda_j$.
- Si $|\det(A)| > 1$ entonces $1 < \rho(A) \leq \|A\|$.

Ejemplo. Si $A = \begin{pmatrix} -1 & 2 & 1 \\ 2 & 3 & -1 \\ 0 & 0 & 1 \end{pmatrix}$, entonces $\|A\|_1 = \max\{3, 5, 3\} = 5$, $\|A\|_\infty = \max\{4, 6, 1\} = 6$ y $\|A\|_2 = 3.8767$. Además $\det(A) = -7 \Rightarrow \rho(A) = 3.8284 > 1$.

Observación. Las normas vectoriales clásicas se relacionan:

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{k} \|x\|_\infty;$$

$$\|x\|_\infty \leq \|x\|_1 \leq k \|x\|_\infty;$$

$$\|x\|_2 \leq \|x\|_1 \leq \sqrt{k} \|x\|_2.$$