Taller # 1

Raices de Funciones

Andrés David Guerrero Durán
Código: 2188138

Modelado Matemático II
Universidad Industrial de Santander

30 de agosto de 2018

¿Cómo se encuentra las raices de una función?. A partir de la solución de una ecuación cuadrática o para funciones polinomiales de bajo grado similares, varios métodos analíticos son utilizados. Sin embargo, para polinomios de alto grado (cinco o más) donde no hay soluciones analíticas generales o para funciones complicadas (o conjuntos de funciones), a menudo es necesario recurrir a **métodos numéricos**.

Los algoritmos de búsqueda de raíces comparten un enfoque muy directo e intuitivo para aproximar las raíces. La estructura general de estos métodos es:

a. Comenzar con una suposición inicial.

b. Calcular el resultado de la suposición.

c. Actualizar la suposición en función del resultado y algunas otras condiciones.

d. Repetir hasta que el método converge a la solución.

Ahora se considera las funciones $f(x) = x^2 - 1$ y $f(x) = tan(x)$ con $x = \pm 1$ y $x = n\pi$ sus raices respectivas, con $n \in \mathcal{Z}$. Como muestra la figura Figura 1

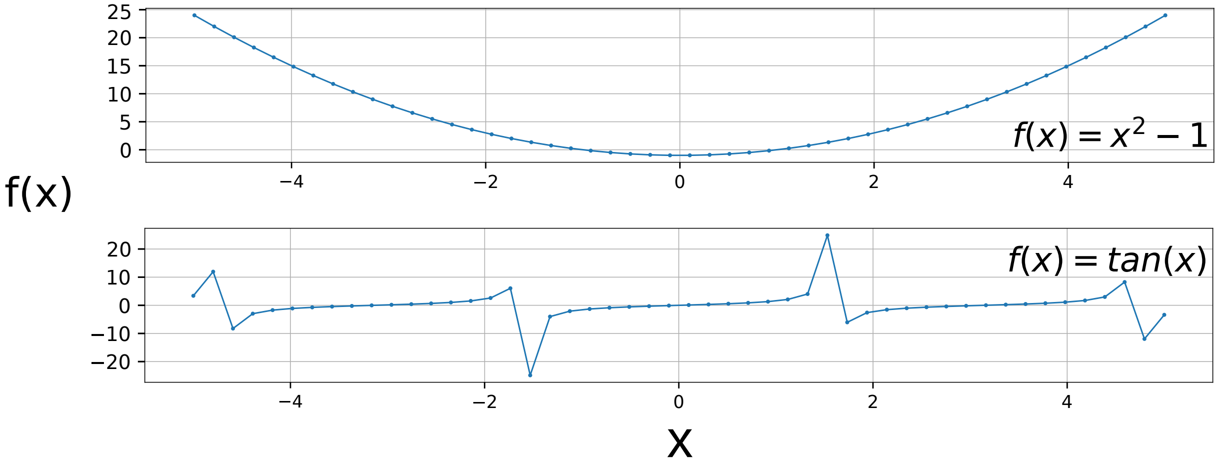
A partir de una tolerancia de error de $1e^{-6}$ se procede a calcular los resultados de convergencia a medida que aumentan el número de iteraciones.

Para la ecuación cuadrática $f(x) = x^2 - 1$ con raices $x = \pm 1$ se hizo una busqueda de la raiz alrededor de los puntos $x = 0$ y $x = 3$

Para la iteración k se calculó el nuevo x hasta converger a la raiz ($x = 1$). De la figura Figura 2 se puede observar que el *Método de Bisection* necesita de aproximadamente 20 iteraciones para encontrar la raiz, sin embargo se acerca más rápido a la solución en la primeras 5 iteraciones a comparación del *Métodos de Brents* y *IQI*.

En la figura Figura 3 el cálculo del punto actual con el punto anterior es evaluado a medida que aumentan la cantidad de iteraciones. Aunque el *Método de Brents*, el *Método de Secante* y el *método IQI* (Inverse Quadratic Interpolation) comienzan en un

Figura 1. Funciones para encontrar sus raices



punto lejos de la raiz, su caida de convergencia es rápida. Sin embargo, el *método de Newton* tiene ventaja en este sentido, por su caida casi lineal con menos iteraciones necesarias.

En la figura Figura 4 se examina el error relativo, es decir, la diferencia entre el punto encontrado en la iteración k y la verdadera raiz($x^* = 1$).

En general para la evaluar estos métodos de busqueda de la raiz debemos revisar los siguientes puntos

- ✓ Tasa de convergencia, es decir, que tan rápido llega a la respuesta.
- ✓ Qué garantias tiene el método de converger.
- ✓ Gasto computacional en cada paso de la iteración.

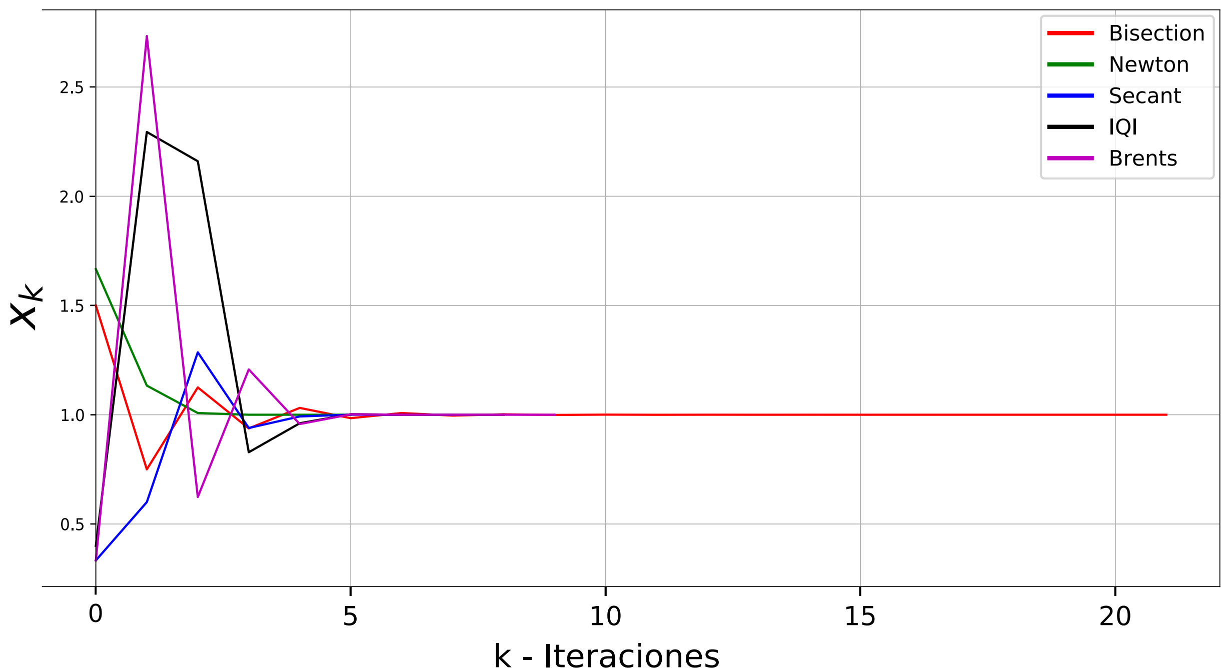
La convergencia se define en términos de la relación de los términos de error sucesivos, o de cuánto disminuye el error por iteración. Si x^* es la raíz verdadera y x_k es la estimación en la iteración k, entonces el error en la iteración k es $\|x_k - x^*\|$ como se mostró en la figura Figura 4, y se dice que converge con la tasa r si (1):

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^r} = \lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C \tag{1}$$

Para una constante finita $C > 0$

- ✓ Si k=1 y $C < 1$, la tasa de convergencia es lineal
- ✓ Si $1 < k < 2$, la tasa de convergencia es superlineal
- ✓ Si k=2, la tasa de convergenica es cuadrática

Figura 2. Punto encontrado vs. Iteraciones



Para ver la comparación entre los métodos con más detalle númerico, la tablas Tabla 1 y Tabla 2 muestran la busqueda de raiz a traves de las iteraciones para la funcion $f(x) = tan(x)$. Además se especifica el tiempo requerido.

La busqueda de las raices se hizo con puntos iniciales cercanos a la raiz $x = \pi$, la función $f(x) = tan(x)$ cruza el eje x cada π. En el caso del IQI los 3 puntos iniciales debian estar cerca a la raiz para lograr llegar a la solución.

Además, en la tablas se puede observar el tiempo requerido para encontrar la raiz. El método de secante obtiene un resultado más rápido. Cabe aclarar que las tablas solo muestran las iteraciones hasta $k = 6$ y para el método de Bisección son requeridas 21 iteraciones.

Tabla 1

	Bisección t = 0.2 ms			**Newton** t = 0.42 ms			**Secante** t = 69 μs		
k	***xk***	***delta***	***ek***	***xk***	***delta***	***ek***	***xk***	***delta***	***ek***
1	3.0	1.0	2	3.5053214495250886	0.49467855047491127	2.50532145	3.3072872417272237	0.6927127582727763	2.30728724
2	3.5	0.5	2.5	3.1728351039874014	0.3324863455376872	2.1728351	3.1903464110633863	0.11694083066383731	2.19034641
3	3.25	0.25	2.25	3.1416129809117157	0.031222122307568576	2.14161298	3.142169701889806	0.04817670917358052	2.14159312
4	3.125	0.125	2.125	3.141592653589802	2.03273219136031e-05	2.14159265	3.141593116271127	0.0005765856186790153	2.14159265
5	3.1875	0.0625	2.1875	3.141592653589793	8.759319515859237e-15	2.14159265	3.1415926535898446	4.6268128217619164e-07	
6	3.15625	0.03125	2.14160156						

Finalmente, se puede concluir:

Bisection es muy robusto y garantiza encontrar una aproximación en un número fijo de iteraciones, pero es muy lento: convergencia lineal. Cada iteración es rápida ya que no calculamos ningún gradiente, pero la convergencia puede ser muy lenta.

Figura 3. Tolerancia vs. iteraciones

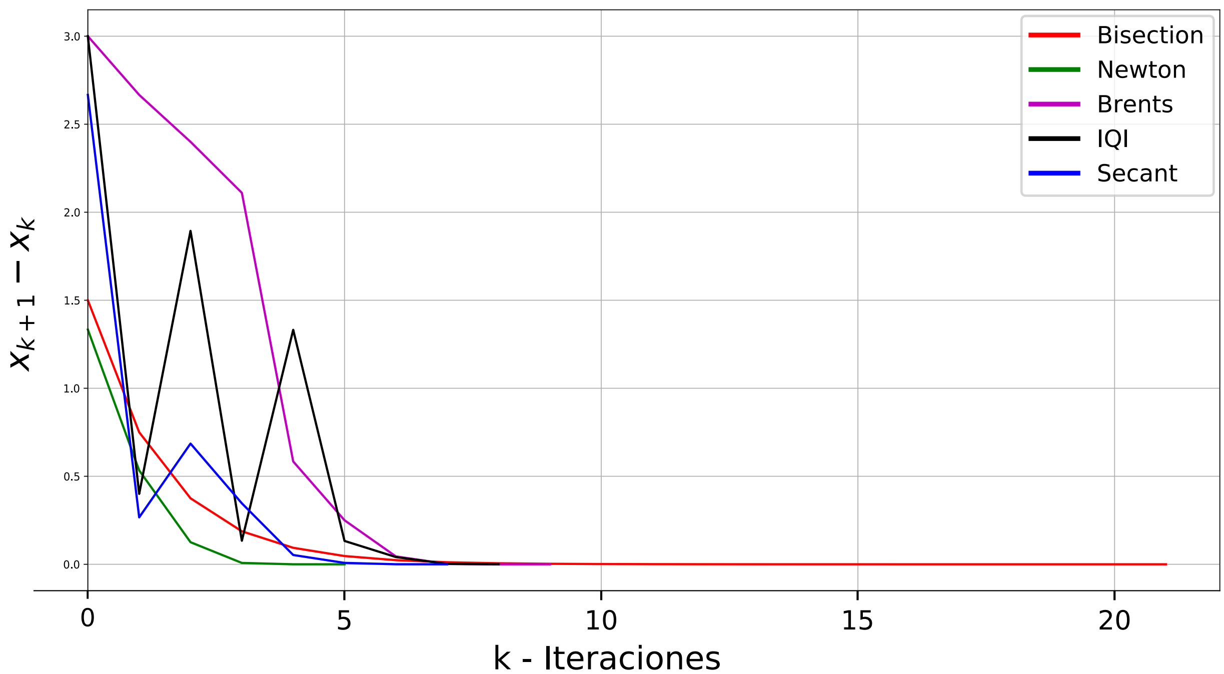


Tabla 2

	IQI t = 0.37 ms			**Brents** t = 1 ms		
k	***xk***	***delta***	***ek***	***xk***	***delta***	***ek***
1	*3.106570271726972*	0.5	2.10657027	3.3072872417272237	2	2.30728724
2	*3.1425579160934696*	1.106570271726972	2.14255792	3.198659987512624	1.3072872417272237	2.19865999
3	3.141585535232105	0.035987644366497484	2.14159265	3.144129940317885	1.1986599875126238	21.14412994
4	3.141592653509604	0.0009723808613646412	2.14159265	3.141626294215345	1.1441299403178848	2.14162629
5	3.141592653589793	7.118277498996406e-06		3.1415926722955487	1.141626294215345	2.14159267
6				3.1415926535899303	1.1415926722955487	2.14159265

Newton converge muy rápido (en forma cuadrática), pero requiere buenas conjeturas iniciales. Como usa la derivada, la función debe ser diferenciable, la derivada debe ser suministrada (o aproximada, lo que puede ser un poco caro y ralentiza la convergencia).

Secante tiene convergencia superlineal, pero es típicamente más rápido que Newton debido a un cálculo relativamente económico en cada iteración. Al igual que Newton, requiere buenas conjeturas iniciales.

IQI este es un método que no se usa directamente para encontrar la raices, de hecho, es parte del método de Brent. Sin embargo, podemos observar en la tabla Tabla 2 este método requiere de menos tiempo para converger.

Brents es esencialmente el método de Bisección aumentado con IQI siempre que el paso sea seguro.

Figura 4. Error relativo vs. iteraciones