Ejemplo. Cálculo de ceros usando el Teorema de Bolzano

Juan Gabriel Gomila, Arnau Mir y Llorenç Valverde

Cálculo de ceros de funciones usando el Teorema de Bolzano

Introducción

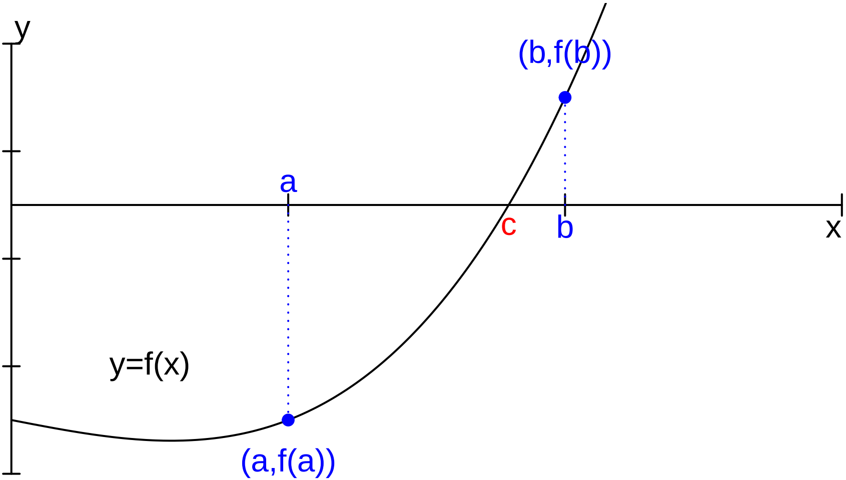
En esta presentación vamos a ver una utilidad del Teorema de Bolzano: cálculo de ceros de funciones.

En primer lugar recordemos su enunciado:

Proposición: Teorema de Bolzano

Sea $f:[a,b]\to\mathbb{R}$ continua tal que $f(a)\cdot f(b)<0$, entonces existe $c\in(a,b)$ tal que $f(c)=0$.

Introducción



Algoritmo para calcular el cero c

Supongamos para fijar ideas que $f(a) < 0$ y $f(b) > 0.$ Si fuese al revés, $f(a) > 0$ y $f(b) < 0.$ Se razonaría de forma parecida.

Vamos a definir dos sucesiones $(a_n)_n$ y $(b_n)_n$ de valores reales tal que $\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = c$ de la forma siguiente:

Algoritmo para calcular el cero c

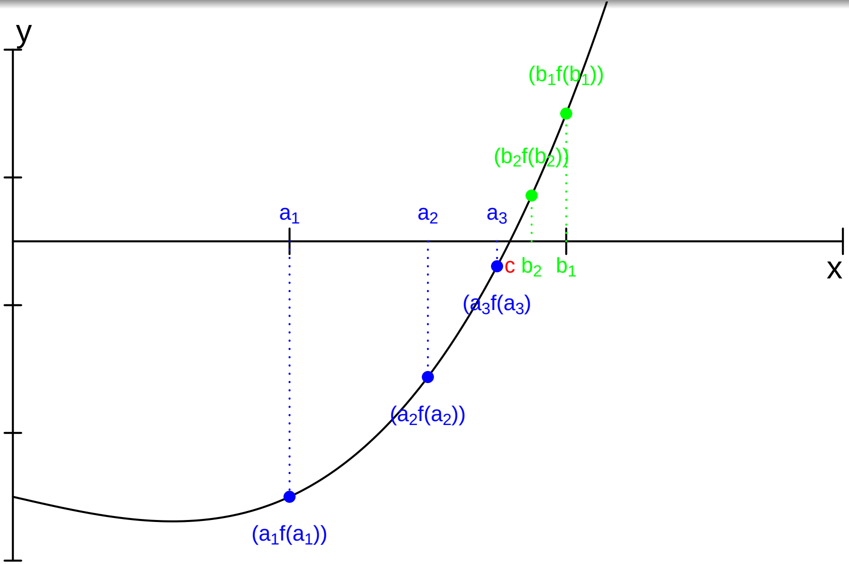
Input : a, b y ϵ con $f(a) < 0$ y $f(b) > 0$
Output : c tal que $|f(c)| < \epsilon$.

$c = \frac{a+b}{2}$

while $|f(c)| \geq \epsilon$ **do**
    if $f(c) < 0$ **then**
        $a = c$
    else
        $b = c$
    end
    $c = \frac{a+b}{2}$
end
return c

Algorithm 1: Algoritmo para calcular el cero de $f(x) = 0$

Algoritmo para calcular el cero c



Ejemplo

Consideremos la función $f(x) = x^3 - x - 4$.

Vamos a crear una tabla de dos columnas: en la primera, vamos a escribir la sucesión $(a_n)_n$ y en la segunda la sucesión $(b_n)_n$.

Nos dicen que $a = a_1 = 1$ con $f(1) = -4 < 0$ y $b = b_1 = 2$ con $f(2) = 2 > 0$.

La primera fila de la tabla será:

a_n	b_n
1	2

Ejemplo

Sea ahora $c = \frac{a_1+b_1}{2} = \frac{1+2}{2} = 1.5.$ El valor de $f(c)$ es $f(1.5) = -2.125.$ Como es negativo, tendremos que $a_2 = 1.5$ y la tabla será:

a_n	b_n
1.0	2
1.5	

Ejemplo

Sea ahora $c = \frac{a_2+b_1}{2} = \frac{1.5+2}{2} = 1.75$. El valor de $f(c)$ es: $f(1.75) = -0.390625$. Como es negativo, tendremos que $a_3 = 1.75$ y la tabla será:

a_n	b_n
1.00	2
1.50	
1.75	

Ejemplo

Sea ahora $c = \frac{a_3+b_1}{2} = \frac{1.75+2}{2} = 1.875$. El valor de $f(c)$ es: $f(1.875) = 0.7167969$. Como es positivo, tendremos que $b_2 = 1.875$ y la tabla será:

a_n	b_n
1.00	2
1.50	1.875
1.75	

Ejemplo

Sea ahora $c = \frac{a_3+b_2}{2} = \frac{1.75+1.875}{2} = 1.8125$. El valor de $f(c)$ es: $f(1.8125) = 0.1418457$. Como es positivo, tendremos que $b_3 = 1.8125$ y la tabla será:

a_n	b_n
1.00	2.0000
1.50	1.8750
1.75	1.8125

Ejemplo

Hagamos el último paso:

Sea ahora $c = \frac{a+b}{2} = \frac{1.75+1.8125}{2} = 1.78125$. El valor de $f(c)$ es: $f(1.78125) = -0.1296082$. Como es negativo, tendremos que $a_4 = 1.78125$ y la tabla será:

a_n	b_n
1.00000	2
1.50000	1.875
1.75000	1.8125
1.78125	

Ejemplo

La precisión de la sucesión a_n es $f(1.78125) = -0.1296082$ y la de b_n es $f(1.8125) = 0.1418457$. Vemos que tenemos poca precisión. Si queremos llegar a una precisión de 0.001, las sucesiones a_n y b_n son las siguientes:

a_n	b_n
1.000000	2
1.500000	1.875
1.750000	1.8125
1.781250	1.796875
1.789062	
1.792969	
1.794922	
1.795898	

Ejemplo

El valor c buscado sería

$$c = \frac{1.7958984 + 1.796875}{2} = \frac{1.7958984 + 1.796875}{2} = 1.7963867,$$

con

$$f(1.7963867) = 5.6264165 \times 10^{-4}.$$

Ejemplo

Para llevar a cabo el ejemplo anterior en `python`, haríamos lo siguiente:

```
def f(x):
 return(x**3-x-4)
a=1.
b=2.
c=(a+b)/2.
epsilon=0.001
while abs(f(c)) >= epsilon:
  if f(c)<0:
    a=c
  else:
    b=c
  c=(a+b)/2
```

Ejemplo

El valor de c sería:

```
print("El cero vale con un error menor que\
{0:.3f}: {1:.7f}".format(epsilon,c))
```

```
## El cero vale con un error menor que0.001: 1.7963867
```