Aplicación del Teorema de Bolzano

Búsqueda de la Raíz de 2

Documento diseñado para el curso de Matemáticas 2, Primavera 2020

Por: Benjamín Vera

En cátedra se vio, entre otros teoremas de continuidad, el Teorema de Bolzano, cuyo enunciado formal el siguiente:

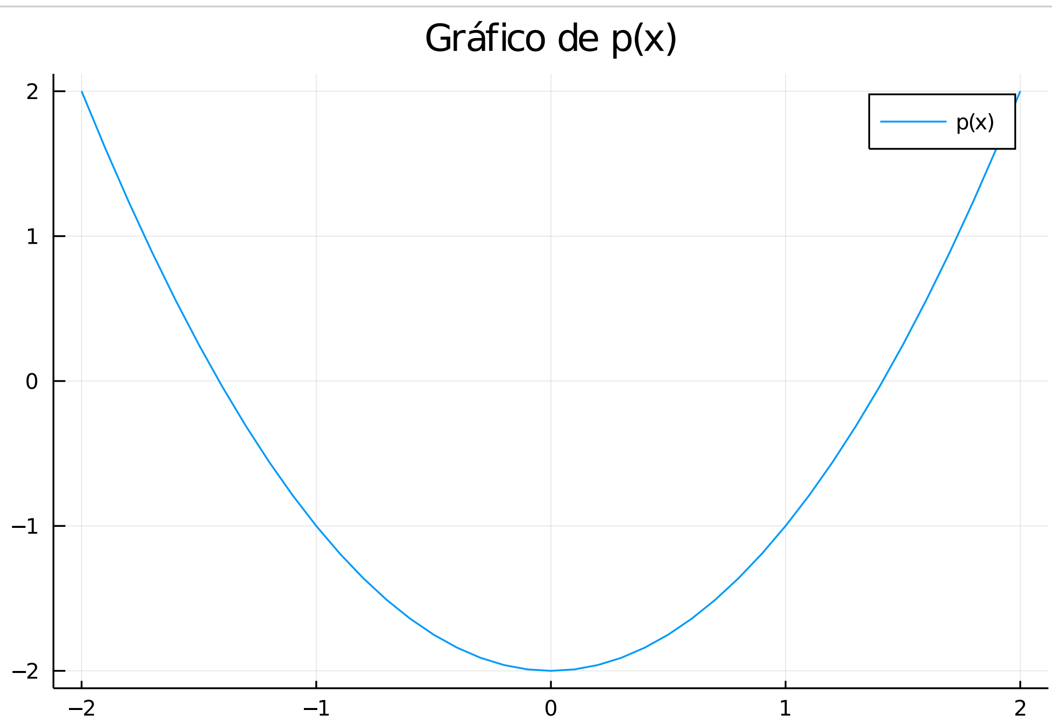
Sea $f:[a,b]\to\mathbb{R}$ una función continua. Entonces, si $f(a)\cdot f(b)<0$, entonces existe $c\in(a,b)$ tal que $f(c)=0$

En este documento ilustraremos una forma computacional de utilizar este teorema para estimar el valor de ciertos números irracionales con grado arbitrario de precisión. Como ejemplo, estimaremos el valor de $\sqrt{2}$. También, al final del documento se mencionarán las limitaciones de este método.

La raíz de 2 corresponde a un número $x\in\mathbb{R}$ que satisface $x^2=2 \iff x^2-2=0$. Por lo tanto, se puede interpretar como una de las dos raíces del polinomio $p(x)=x^2-2$. En la siguiente celda, utilizamos el lenguaje Julia (1.5.2) para graficar el polinomio p.

In [1]:
```
using Plots
# plotly()
x = -2:0.1:2
p = x.^2 .- 2
plot(x, p, label="p(x)")
title!("Gráfico de p(x)")
```

Out[1]:



Formalmente, para demostrar la existencia de la raíz de 2, diríamos que el polinomio $p(x)$ corresponde a una función continua en todo $\mathbb{R}$, en particular en $[1, 2]$ y por lo tanto, sabiendo que $p(1) = 1^2 - 2 = -1$ y que $p(2) = 2^2 - 2 = 2.$ Ya que ambos números tienen distinto signo (o, según el enunciado formal de arriba, que su producto es negativo), el T.B nos permite asegurar la existencia de un $c \in [1, 2]$ tal que $p(c) = 0 \iff c^2 - 2 = 0 \iff c^2 = 2$ ■

Esto ya lo sabemos. Ahora, resta encontrar un método para aproximar este número.

Método de aproximación

La idea básica de aproximar utilizando el teorema de bolzano consiste en implementar siguiente algoritmo:

1. Ubicar un intervalo inicial $[a, b]$ en que se sepa que el punto c se encuentra y es único (esto es, deberíamos asegurar que no va a haber otra raíz en este intervalo, ya que de lo contrario el algoritmo falla).
2. Tomar d como el punto medio entre a y b. Ya que el c que buscamos es único, solo uno de los productos $f(a) \cdot f(d)$ y $f(d) \cdot f(b)$ va a ser negativo.
3. Si, por ejemplo, el primer producto es negativo, hemos reducido a la mitad nuestro intervalo de búsqueda, ahora repetimos los pasos con $[a, d]$ y se itera hasta obtener la precisión deseada.

A continuación, implementamos el método señalado en Julia:

In [18]:
```julia
# Definimos para Julia la función con la que vamos a trabajar
p = x -> x^2 - 2

a = 1
b = 2  # Originalmente, sabemos que la raíz se ubica entre 1 y 2
epsilon = 0.00001  # Aproximaremos la raíz de 2 dentro de un intervalo de largo 0.00001.
Cambiando este número lograremos más
# precisión. Sin embargo, el loop que estamos a punto de construír tendrá que correr más
veces.
n = 0  # Vamos a contar las iteraciones también.
while b-a > epsilon
    d = (a+b)/2  # Construyo el punto medio
    if p(d) == 0  # Si en d el polinomio vale cero, significa que por suerte, hemos enco
ntrado el punto que buscamos
        println("Lo encontré! La solución es exactamente $d")
    elseif p(a)*p(d) < 0  # Si p(a) y p(d) tienen distinto signo...
        b = d  # Redefino el extremo superior del intervalo
    else  # Si no se cumple, es porque el producto negativo es p(d)*p(a)
        a = d  # Y en ese caso debemos redefinir el extremo inferior
    n += 1  # Aumento en 1 el valor de n para contar las iteraciones
    end
end
# Al final de todo el proceso, indiquemos nuestros resultados:
println("Tras $n iteraciones, el número buscado se ubica entre $a y $b")
```

```
Tras 7 iteraciones, el número buscado se ubica entre 1.4142074584960938 y 1.414215087890
625
```

Vemos que no es difícil lograr grandes niveles de precisión con este algoritmo. En el sentido de que el número de iteraciones requeridas no escala tan rápido conforme aumentamos la precisión buscada. Se pueden introducir modificaciones a este algoritmo para manipular directamente la **cantidad de dígitos** en lugar del largo de un intervalo. Para esto, en lugar de tomar el punto medio, lo que buscamos es subdividir el intervalo $[a, b]$ en diez sub-intervalos de igual longitud. Con esto, buscar cuál de estos intervalos cumple la condición de bolzano ($f(b) \cdot f(a) < 0$) corresponde directamente a tener un dígito más de precisión.

Limitaciones

Si bien este método no falla al estimar $\sqrt{2}$, no todo número irracional puede ser aproximado de esta forma. El candidato que quizá más quisiéramos aproximar mediante este método es el número π. Pero... ¿Qué ecuación conocemos que tenga como resultado π?

Una opción sería utilizar trigonometría. Ya que $\cos(\pi/2) = 0$, la podríamos buscar como uno de los ceros de la función $g(x) = \cos(x/2)$, función que sin duda es continua en todos los reales. Esto es viable computacionalmente, pero es técnicamente circular ya que para esto tendríamos que ser capaces de calcular el valor numérico de la función $\cos x$ en distintos puntos de su dominio. ¡¡Pero para hacer esto hace falta calcular números irracionales que son precisamente el problema que estamos tratando de resolver!! Este problema conceptual no lo encontramos antes ya que estábamos evaluando valores en un polinomio, el cual solamente consiste en sumas y productos, que son posibles de hacer a mano (aunque tediosas, que es la razón por la que lo estamos programando)

Por lo tanto, si bien podríamos hacer trampa con Julia y estimar π mediante la ecuación $\cos(x/2) = 0$, esto sería equivalente a simplemente escribir el valor de π en una calculadora. Así que lo que en realidad buscamos es un **polinomio** p(x) una de cuyas raíces sea π. Y se ha demostrado (¡no hace mucho!) que dicho polinomio no puede existir.

Un poco de terminología matemática: Decimos que un número c es *algebraico* si existe polinomio $p(x)$ a coeficientes enteros tal que $p(c) = 0$, y con ello se define que un número es *trascendente* si **no** es algebraico. En suma, lo que se ha demostrado (y la demostración escapa bastante del alcance del curso) es que π es un número trascendente.

En conclusión, el método de Bolzano falla para aproximar números trascendentes, uno de los cuales es π. Por lo tanto, se requiere de otras herramientas más avanzadas para resolver este problema. Con el avance del curso a lo largo del semestre exploraremos algunas de ellas.