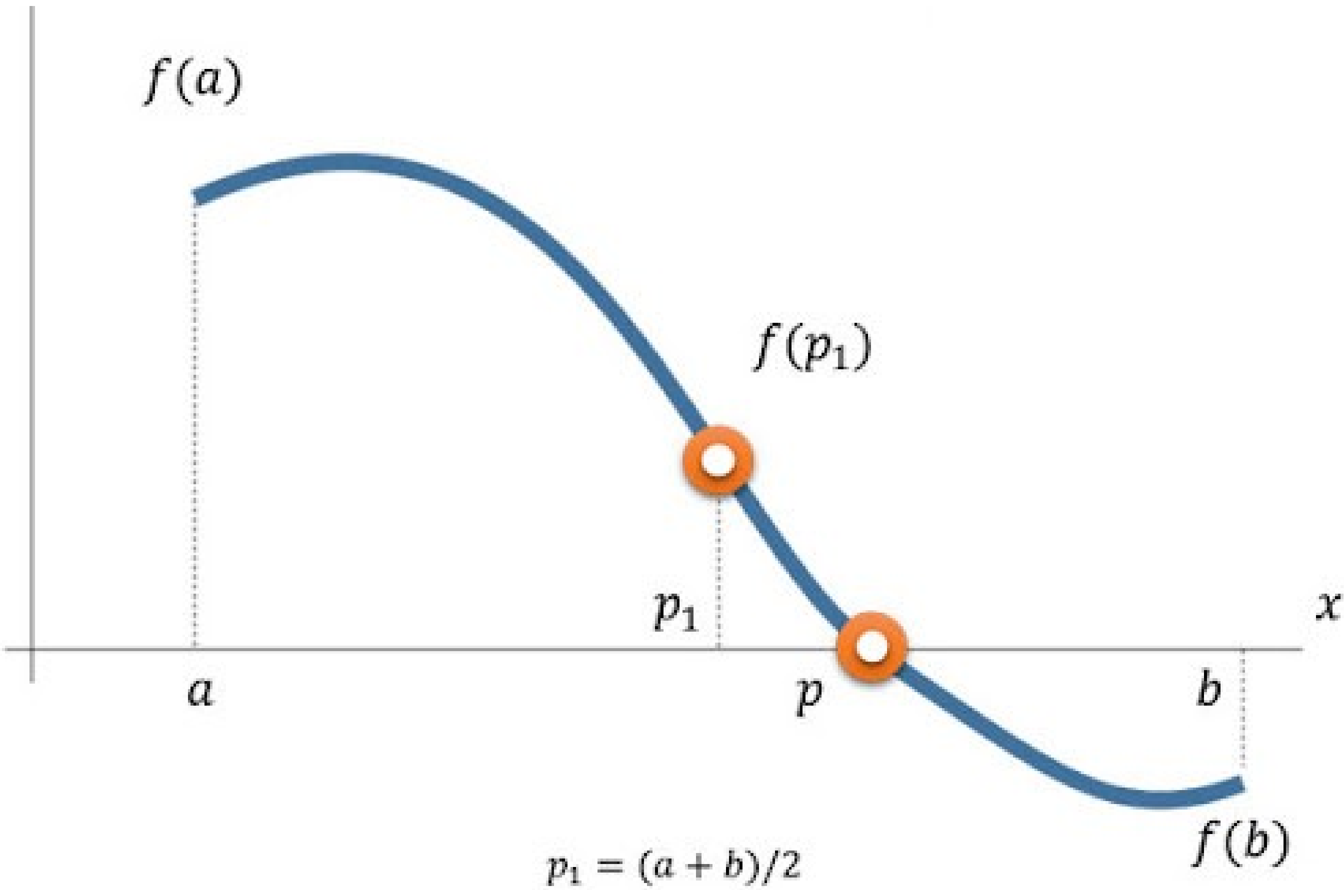
Resolución numérica de ecuaciones: método de bisección

Se presenta aqui un método sencillo,basado en el **Teorema de Bolzano**, que permite, en determinadas circunstancias,calcular la solución de una ecuación. Hay que comenzar por decir que cualquier ecuación en una variable se puede escribir (y no de manera única) en la forma de una equivalente (es decir, que tiene las mismas soluciones) pero con segundo miembro nulo, es decir: $f(x) = 0$.

Dada $f : [a, b] \subset \mathbb{R} \longrightarrow \mathbb{R}$, continua, se plantea el problema de encontrar una solución de la ecuación: $f(x) = 0$

Desde el punto de vista geométrico, esto, en $[a, b]$, un punto de corte de la gráfica de la función $y = f(x)$ con el eje de las abscisas. Tal como se muestra en la siguiente figura:



Los métodos de aproximación de raices de ecuaciones necesitan conocer, o bien un intervalo que contenga sólo una raíz, o bien un punto inicial que esté suficientemente cerca de ella. Por tanto, como paso previo a la aplicación de un método de aproximación, es necesario localizar la raíz, es decir encontrar un intervalo que la contenga y separar la raíz, es decir encontrar un intervalo que sólo contenga dicha raíz. Esto se hace por métodos analíticos, gráficos y, en algunos casos, empíricos.

Los métodos para aproximar raices de ecuaciones son, en general iterativos, es decir consisten en construir una sucesión de valores $x_1, x_2, x_3, x_4 \ldots$ mediante una realación de recurrencia, esto es, se calcula uno de ellos a partir del anterior: $x_1 \to x_2 \to x_3 \to x_4$, etc.

Cuando la sucesión de valores $x_1, x_2, x_3 \ldots$ tiende hacia la raíz α de $f(x)$ (es decir se acerca cada más a ella, tanto como se quiera: $\lim_{n\to\infty} x_n$), se dice que el método iterativo **es convergente**.

En general, el método de bisección consiste en lo siguiente:

1. Subdividir en dos partes el intervalo en que se sabe la función cambia de signo y tiene una sóla raíz.
2. Averiguar,utilizando el teorema de Bolzano, en el cual de las dos mitades se encuentra la raíz y descartar la otra mitad del intervalo.
3. Reiniciar este proceso con el subintervalo elegido.
4. Continuar con este proceso hasta que el subintervalo elegido tenga una longitud lo suficientemente pequeña como para que cualquiera de sus puntos sea una aproximación aceptable de la solución. La elecccioón óptima como aproximación es, entonces, el punto medio del subintervalo.

Implemetación del código

In [34]:
```
function f(x)
    return cos(x)-x^3
end
```

Out[34]:
```
f (generic function with 1 method)
```

In [35]:
```
f(2)
```

Out[35]:
```
-8.416146836547142
```

In [36]:
```
f(0)
```

Out[36]:
```
1.0
```

In [83]:

```
function f(x)
    return cos(x)-x^3
end

function biseccion(f,a,b,err)
    l1 = a
    l = b
    k = 0
    if f(a)*f(b) > 0
        println("La función no cambia de signo")
    end

    while abs(l1-l)>err
        l1 = l
        l = (a+b)/2
        if f(a)*f(l) < 0   #Cambia de signo en [a,l]
            b = l
        end

        if f(l)f(b) < 0 #Cambia de signo en [l,b]
            a = l
        end
        println("El intervalo es [$a , $b]")
        k = k+1

    end
    println("$k iteraciones")
    print("x", "=",l," ", "con: ", k," ", "iteraciones es una buena aproximaci
on")
end
```

Out[83]:

```
biseccion (generic function with 2 methods)
```

In [84]:
```
biseccion(f,0,π,10^-6)
```

```
El intervalo es [0 , 1.5707963267948966]
El intervalo es [0.7853981633974483 , 1.5707963267948966]
El intervalo es [0.7853981633974483 , 1.1780972450961724]
El intervalo es [0.7853981633974483 , 0.9817477042468103]
El intervalo es [0.7853981633974483 , 0.8835729338221293]
El intervalo es [0.8344855486097889 , 0.8835729338221293]
El intervalo es [0.859029241215959 , 0.8835729338221293]
El intervalo es [0.859029241215959 , 0.8713010875190441]
El intervalo es [0.8651651643675016 , 0.8713010875190441]
El intervalo es [0.8651651643675016 , 0.8682331259432728]
El intervalo es [0.8651651643675016 , 0.8666991451553872]
El intervalo es [0.8651651643675016 , 0.8659321547614444]
El intervalo es [0.8651651643675016 , 0.865548659564473]
El intervalo es [0.8653569119659873 , 0.865548659564473]
El intervalo es [0.8654527857652301 , 0.865548659564473]
El intervalo es [0.8654527857652301 , 0.8655007226648516]
El intervalo es [0.8654527857652301 , 0.8654767542150408]
El intervalo es [0.8654647699901354 , 0.8654767542150408]
El intervalo es [0.8654707621025881 , 0.8654767542150408]
El intervalo es [0.8654737581588144 , 0.8654767542150408]
El intervalo es [0.8654737581588144 , 0.8654752561869277]
El intervalo es [0.8654737581588144 , 0.8654745071728711]
22 iteraciones
x=0.8654745071728711 con: 22 iteraciones es una buena aproximacion
```

In [ ]: