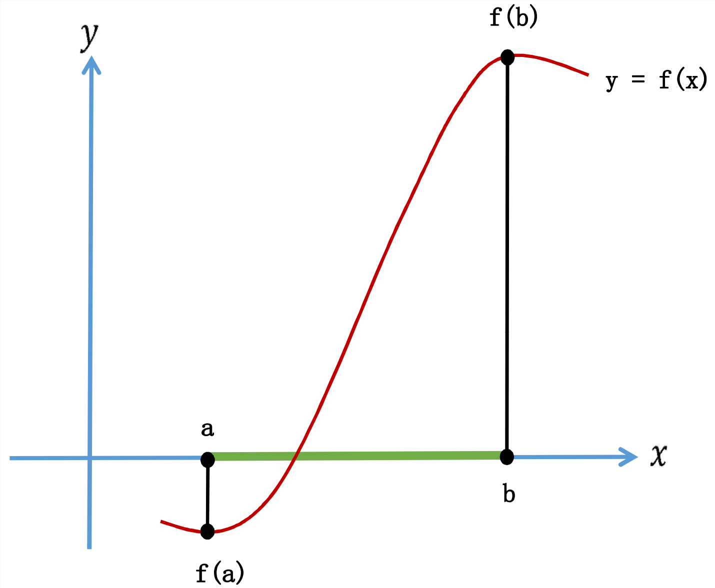
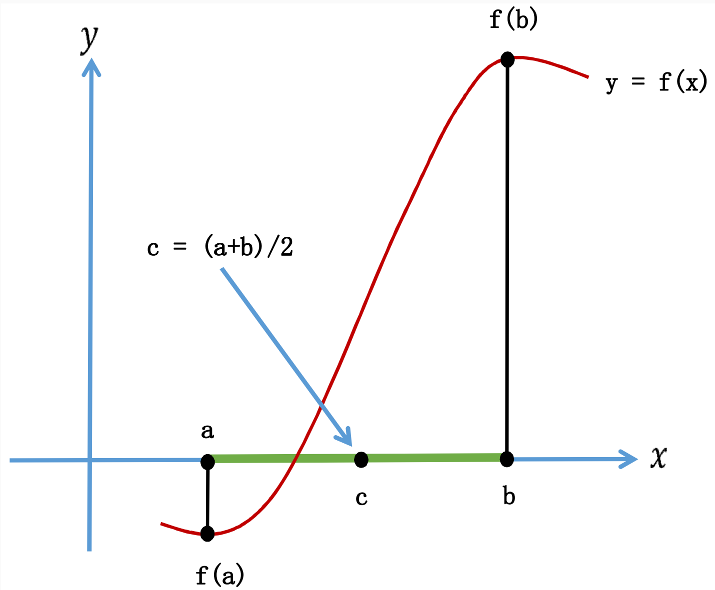
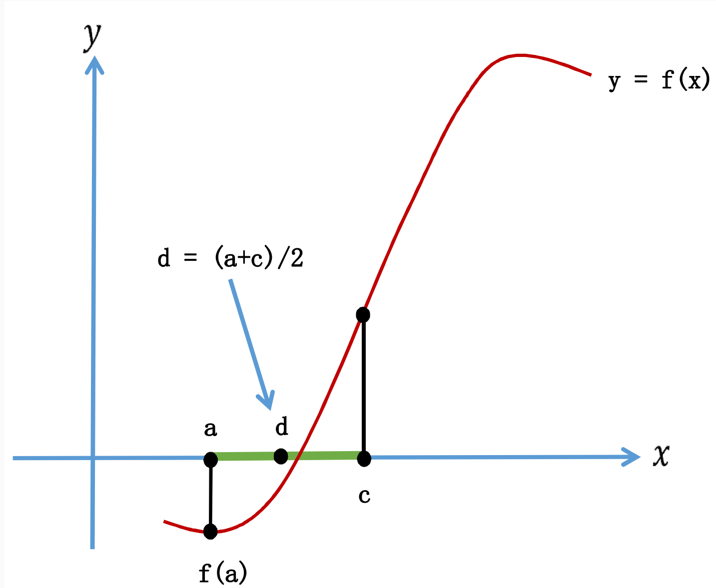
Bisection Method



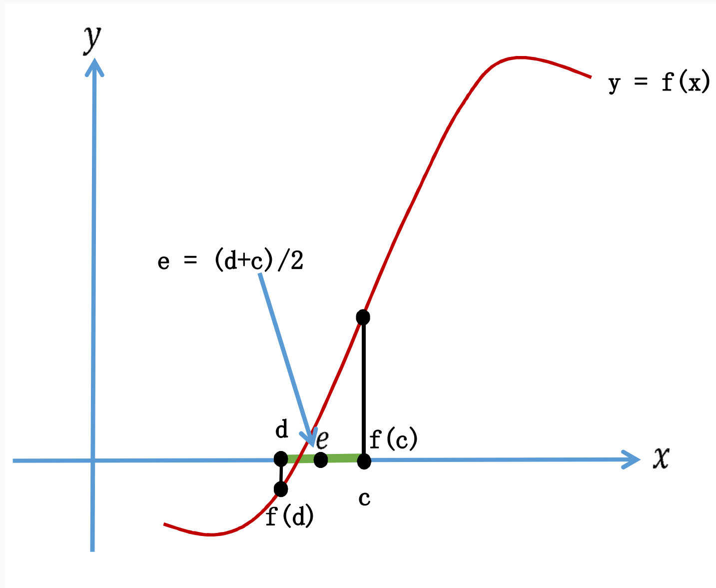
Bisection Method



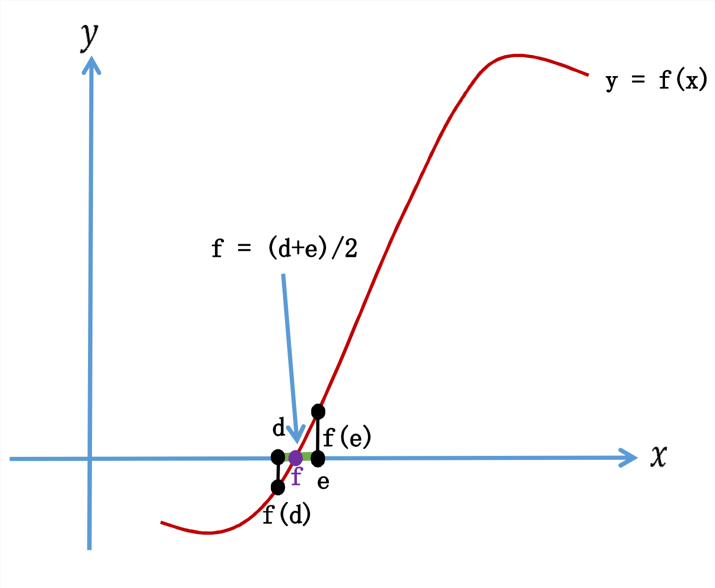
Bisection Method



Bisection Method



Bisection Method



Overview and Assumptions

Bisection Method is based on the Intermediate Value Theorem.

Intermediate Value Theorem (IVT): Suppose a continuous function f, defined on $[a, b]$ is given with $f(a)$ and $f(b)$ of opposite sign then by the IVT, there exists a point $c \in (a, b)$ for which $f(c) = 0$.

Main Assumptions Bisection method may be applied to find the approximate root of a function $f(x)$ satisfying following

- $f(x)$ is a continuous function defined on the interval $[a, b]$, with $f(a)$ and $f(b)$ of opposite sign.
- Although the procedure will work when there is more than one root in the interval (a, b), we assume for simplicity that the root in this interval is unique.

Computational Steps

The method calls for a repeated halving (or bisecting) of subintervals of $[a, b]$ and, at each step, locating the half containing c.

To begin, set $a_1 = a$ and $b_1 = b$, and let $c_1 = \dfrac{a_1 + b_1}{2}$

- If $f(c_1) = 0$, then $c = c_1$, and we are done.
- If $f(c_1) \neq 0$, then $f(c_1)$ has the same sign as either $f(a_1)$ or $f(b_1)$.
  - If $f(c_1)$ and $f(a_1)$ have the same signs, $c \in (c_1, b_1)$. Set $a_2 = c_1, f(a_2) = f(c_1)$ and $b_2 = b_1, f(b_2) = f(b_1)$.
  - If $f(c_1)$ and $f(a_1)$ have opposite signs, $c \in (a_1, c_1)$. Set $a_2 = a_1, f(a_2) = f(a_1)$ and $b_2 = c_1, f(b_2) = f(c_1)$.

Then re-apply the process to the interval $[a_2, b_2]$, etc. until the termination criteria is satisfied.

Bisection Method Pseudo Code

Require: $a_1,\ b_1,\ f(a_1) * f(b_1) < 0,\ \epsilon > 0$

$k = 1, MaxIter = ceil\left(1 + \dfrac{\log((b_1 - a_1)/\epsilon)}{\log 2}\right);$

while $(k \leq MaxIter)$ **do** $\{$

    $c_k = \dfrac{a_k + b_k}{2};$

    if $(f(a_k) * f(c_k) < 0)$ **then** $\{$

        $a_{k+1} = a_k, b_{k+1} = c_k;$

        $f(a_{k+1}) = f(a_k), f(b_{k+1}) = f(c_k);$

    $\}$

    if $(f(b_k) * f(c_k) < 0)$ **then** $\{$

        $a_{k+1} = c_k, b_{k+1} = b_k;$

        $f(a_{k+1}) = f(c_k), f(b_{k+1}) = f(b_k);$

    $\}$

    $k = k + 1;$

$\}$

$c_{k+1} = \dfrac{a_{k+1} + b_{k+1}}{2}$ is the root.

Stopping Criteria

There are several ways to set a termination criteria for the bisection iterations. One can select a tolerance $\epsilon > 0$ and generate $c_1, c_2, \ldots, c_n$ until one of the following conditions is met:

$$|c_n - c_{n-1}| < \epsilon$$

or,

$$\frac{|c_n - c_{n-1}|}{|c_n|} < \epsilon$$

or,

$$|f(c_n)| < \epsilon$$

Minimum iterations required for given tolerance

Given initial interval $[a_1, b_1]$, It is easier find minimum number of iterations required for a given tolerance level ϵ allowed in the final interval of uncertainty $[a_n, b_n]$. Since,

$$\begin{aligned} b_n - a_n &= \frac{1}{2}(b_{n-1} - a_{n-1}) = \frac{1}{2}\left(\frac{1}{2}(b_{n-2} - a_{n-2})\right) \\ &= \frac{1}{2^2}(b_{n-2} - a_{n-2}) = \frac{1}{2}\left(\frac{1}{2}(b_{n-3} - a_{n-3})\right) \\ &= \frac{1}{2^3}(b_{n-3} - a_{n-3}) = \frac{1}{2^{(n-1)}}(b_1 - a_1) \end{aligned}$$

Therefore, $b_n - a_n < \epsilon \Rightarrow \frac{1}{2^{(n-1)}}(b_1 - a_1) < \epsilon$

$$\Rightarrow n > 1 + \frac{\log(b_1 - a_1) - \log \epsilon}{\log 2}$$

Example

Question: Show that $f(x) = x^3 + 4x^2 - 10 = 0$ has a root in $[1, 2]$ and use the Bisection method to determine an approximation to the root that is accurate to at least within 10^{-2}.

Solution: Since $f(x)$ is continuous (why?) and $f(1) = -5$ and $f(2) = 14$, by IVT $f(x)$ a root in $[1, 2]$.

Set $a_1 = 1$ and $b_1 = 2$

$\Rightarrow c_1 = \frac{a_1 + b_1}{2} = \frac{1+2}{2} = 1.5$ and $f(1.5) = 2.375 > 0$.

Set $a_2 = a_1 = 1$ and $b_2 = c_1 = 1.5$.

$\Rightarrow c_2 = \frac{a_2 + b_2}{2} = \frac{1+1.5}{2} = 1.25$ with $f(c_2) = -1.7969 < 0$.

Set $a_3 = c_2 = 1.25$ and $b_3 = 1.5$.

$\Rightarrow c_3 = \frac{a_3 + b_3}{2} = \frac{1.25+1.5}{2} = 1.375$ with $f(c_3) = 0.1621 > 0$.

Example

Here $a_1 = 1, b_1 = 5$ and $\epsilon = 10^{-2}$.

$\Rightarrow n > 1 + \frac{\log((b_1 - a_1)/\epsilon)}{\log 2} \Rightarrow n > 1 + \frac{\log((5-1)/10^{-2})}{\log 2} = 1 + \frac{\log 400}{\log 2}$

$\Rightarrow n > 1 + 8.64 \Rightarrow n > 9.64$. Hence $n = 10$ iterations would be sufficient to achieve the required accuracy.

iter	a_i	b_i	c_i	$f(a_i)$	$f(b_i)$	$f(c_i)$
1	1.0000	1.5000	1.2500	-5.0000	2.3750	-1.7969
2	1.2500	1.5000	1.3750	-1.7969	2.3750	0.1621
3	1.2500	1.3750	1.3125	-1.7969	0.1621	-0.8484
4	1.3125	1.3750	1.3438	-0.8484	0.1621	-0.3510
5	1.3438	1.3750	1.3594	-0.3510	0.1621	-0.0964
6	1.3594	1.3750	1.3672	-0.0964	0.1621	0.0324
7	1.3594	1.3672	1.3633	-0.0964	0.0324	-0.0321
8	1.3633	1.3672	1.3652	-0.0321	0.0324	0.0001
9	1.3633	1.3652	1.3643	-0.0321	0.0001	-0.0160
10	1.3643	1.3652	1.3647	-0.0160	0.0001	-0.0080