Lecture-02: Bisection method and its application

One of the first numerical methods developed to find the root of a nonlinear equation $f(x)=0$ was the bisection method (also called *binary-search* method). The method is based on the following theorem.

Theorem

- An equation $f(x)=0$, where $f(x)$ is a real continuous function, has at least one root between x_l and x_u if $f(x_l)f(x_u) < 0$ (See Figure 1).

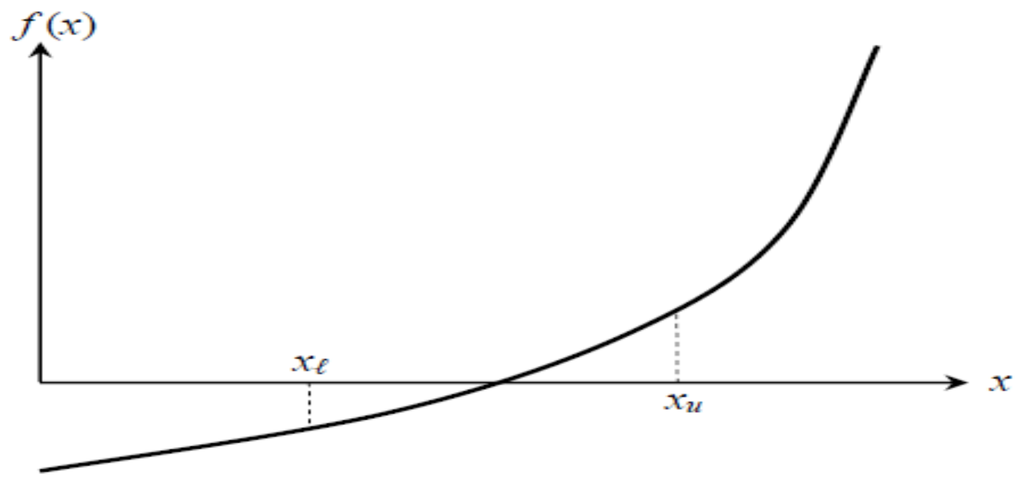

Figure 1 At least one root exists between the two points if the function is real, continuous, and changes sign.

Continued…

- Note that if $f(x_l)f(x_u) > 0$, there may or may not be any root between x_l and x_u (Figures 2 and 3)

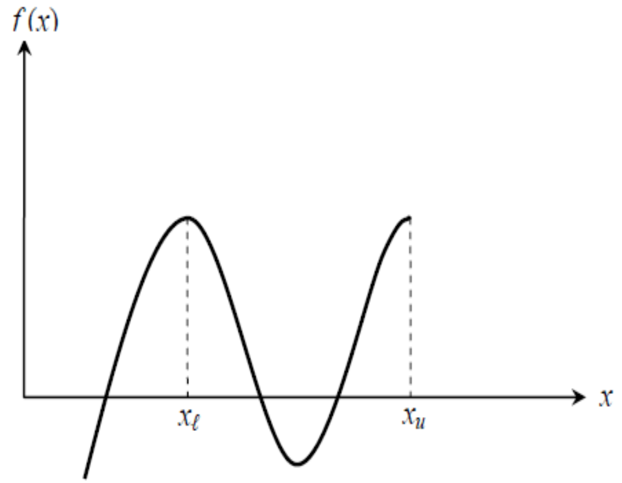

Figure 2 If the function $f(x)$ does not change sign between the two points, roots of the equation $f(x)=0$ may still exist between the two points.

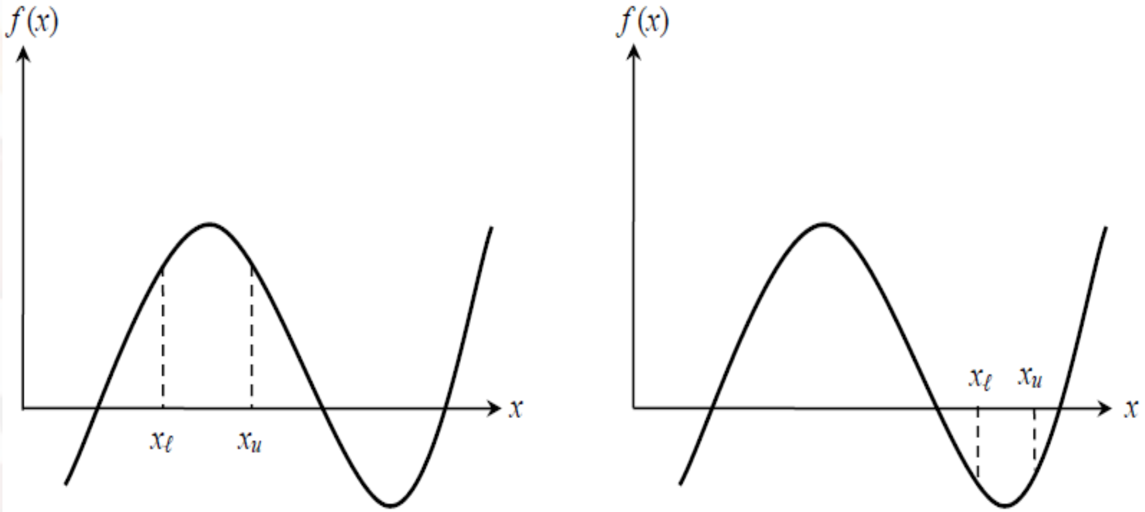

Figure 3 If the function $f(x)$ does not change sign between two points, there may not be any roots for the equation $f(x)=0$ between the two points.

Continued…

- If $f(x_l)f(x_u) < 0$, then there may be more than one root between x_l and x_u (Figure 4). So the theorem only guarantees one root between x_l and x_u.

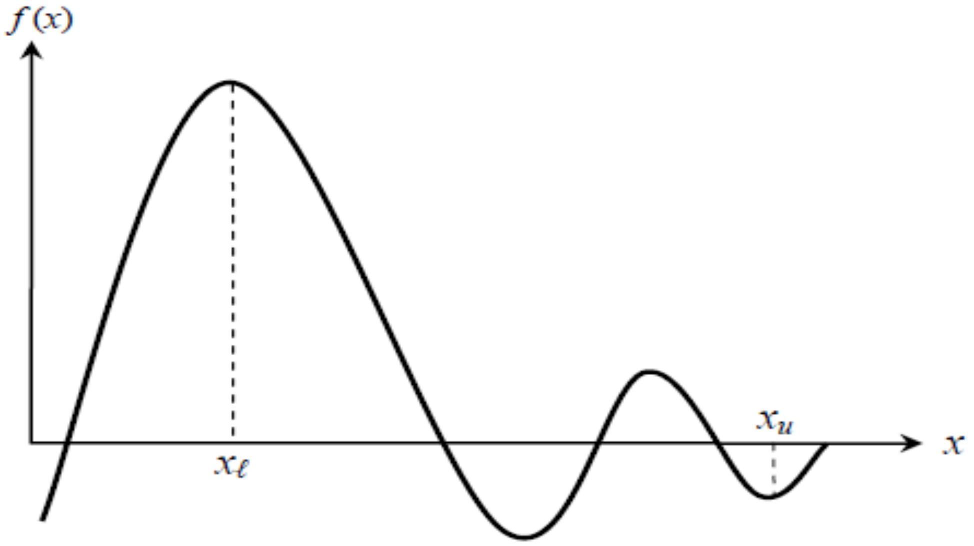

Figure 4 If the function $f(x)$ changes sign between the two points, more than one root for the equation $f(x) = 0$ may exist between the two points.

Algorithm for the bisection method

The steps to apply the bisection method to find the root of the equation are $f(x) = 0$.

1. Choose x_ℓ and x_u as two guesses for the root such that $f(x_\ell)f(x_u) < 0$, or in other words, $f(x)$ changes sign between x_ℓ and x_u.

2. Estimate the root, x_m, of the equation $f(x) = 0$ as the mid-point between x_ℓ and x_u as

$$x_m = \frac{x_\ell + x_u}{2}$$

3. Now check the following
   a) If $f(x_\ell)f(x_m) < 0$, then the root lies between x_ℓ and x_m; then $x_\ell = x_\ell$ and $x_u = x_m$.
   b) If $f(x_\ell)f(x_m) > 0$, then the root lies between x_m and x_u; then $x_\ell = x_m$ and $x_u = x_u$.
   c) If $f(x_\ell)f(x_m) = 0$; then the root is x_m. Stop the algorithm if this is true.

Continued…

4. Find the new estimate of the root

$$x_m = \frac{x_\ell + x_u}{2}$$

Find the absolute relative approximate error as

$$|\in_a| = \left| \frac{x_m^{new} - x_m^{old}}{x_m^{new}} \right| \times 100$$

where

x_m^{new} = estimated root from present iteration

x_m^{old} = estimated root from previous iteration

5. Compare the absolute relative approximate error $|\in_a|$ with the pre-specified relative error tolerance $\in_s$. If $|\in_a| > \in_s$, then go to Step 3, else stop the algorithm. Note one should also check whether the number of iterations is more than the maximum number of iterations allowed. If so, one needs to terminate the algorithm and notify the user about it.

Application:

The equation that gives the depth x to which the ball is submerged under water is given by

$$x^3 - 0.165x^2 + 3.993\times10^{-4} = 0$$

Use the bisection method of finding roots of equations to find the depth x to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation. Find the absolute relative approximate error at the end of each iteration, and the number of significant digits at least correct at the end of each iteration.

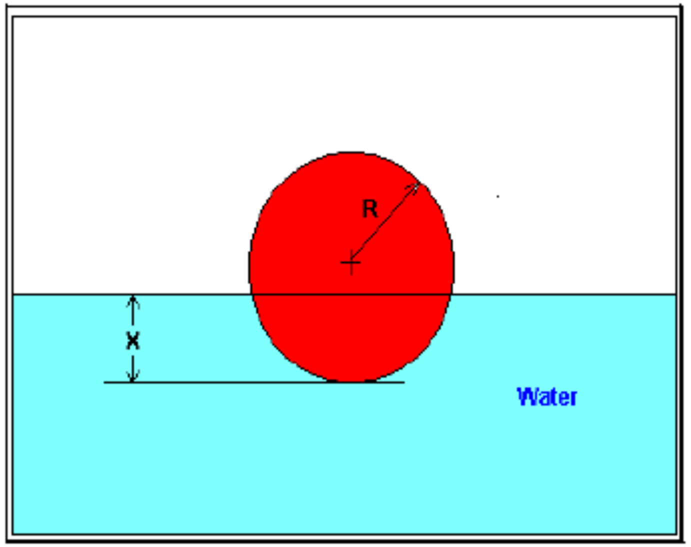

Figure 5 Floating ball problem.

Continued...

Solution

From the physics of the problem, the ball would be submerged between $x=0$ and $x=2R$, where

$$R = \text{radius of the ball},$$

that is

$$0 \le x \le 2R$$
$$0 \le x \le 2(0.055)$$
$$0 \le x \le 0.11$$

Lets us assume

$$x_\ell = 0,\ x_u = 0.11$$

Check if the function changes sign between x_ℓ and x_u.

$$f(x_\ell) = f(0) = (0)^3 - 0.165(0)^2 + 3.993\times10^{-4} = 3.993\times10^{-4}$$
$$f(x_u) = f(0.11) = (0.11)^3 - 0.165(0.11)^2 + 3.993\times10^{-4} = -2.662\times10^{-4}$$

Hence

$$f(x_\ell)f(x_u) = f(0)f(0.11) = (3.993\times10^{-4})(-2.662\times10^{-4}) < 0$$

So there is at least one root between x_ℓ and x_u, that is between 0 and 0.11.

Continued...

Iteration 1

The estimate of the root is

$$x_m = \frac{x_\ell + x_u}{2}$$

$$= \frac{0 + 0.11}{2}$$

$$= 0.055$$

$$f(x_m) = f(0.055) = (0.055)^3 - 0.165(0.055)^2 + 3.993\times10^{-4} = 6.655\times10^{-5}$$

$$f(x_\ell)f(x_m) = f(0)f(0.055) = \left(3.993\times10^{-4}\right)\left(6.655\times10^{-4}\right) > 0$$

Hence the root is bracketed between x_m and x_u, that is, between 0.055 and 0.11. So, the lower and upper limit of the new bracket is

$$x_\ell = 0.055,\ x_u = 0.11$$

At this point, the absolute relative approximate error $|\in_a|$ cannot be calculated as we do not have a previous approximation.

Continued…

Iteration 2

The estimate of the root is

$$x_m = \frac{x_\ell + x_u}{2}$$
$$= \frac{0.055 + 0.11}{2}$$
$$= 0.0825$$

$$f(x_m) = f(0.0825) = (0.0825)^3 - 0.165(0.0825)^2 + 3.993 \times 10^{-4} = -1.622 \times 10^{-4}$$

$$f(x_\ell)f(x_m) = f(0.055)f(0.0825) = \left(6.655 \times 10^{-5}\right) \times \left(-1.622 \times 10^{-4}\right) < 0$$

Hence, the root is bracketed between x_ℓ and x_m, that is, between 0.055 and 0.0825. So the lower and upper limit of the new bracket is

$$x_\ell = 0.055, x_u = 0.0825$$

The absolute relative approximate error $|\in_a|$ at the end of Iteration 2 is

$$|\in_a| = \left| \frac{x_m^{new} - x_m^{old}}{x_m^{new}} \right| \times 100$$
$$= \left| \frac{0.0825 - 0.055}{0.0825} \right| \times 100$$
$$= 33.33\%$$

None of the significant digits are at least correct in the estimated root $x_m = 0.0825$ of because the absolute relative approximate error is greater than 5%.

Continued…

Iteration 3

$$x_m = \frac{x_\ell + x_u}{2}$$
$$= \frac{0.055 + 0.0825}{2}$$
$$= 0.06875$$

$$f(x_m) = f(0.06875) = (0.06875)^3 - 0.165(0.06875)^2 + 3.993\times 10^{-4} = -5.563\times 10^{-5}$$

$$f(x_\ell)f(x_m) = f(0.055)f(0.06875) = (6.655\times 10^5)\times(-5.563\times 10^{-5}) < 0$$

Hence, the root is bracketed between x_ℓ and x_m, that is, between 0.055 and 0.06875. So the lower and upper limit of the new bracket is

$$x_\ell = 0.055,\ x_u = 0.06875$$

The absolute relative approximate error $|\epsilon_a|$ at the ends of Iteration 3 is

$$|\epsilon_a| = \left|\frac{x_m^{\text{new}} - x_m^{\text{old}}}{x_m^{\text{new}}}\right|\times 100$$
$$= \left|\frac{0.06875 - 0.0825}{0.06875}\right|\times 100$$
$$= 20\%$$

Continued…

Still none of the significant digits are at least correct in the estimated root of the equation as the absolute relative approximate error is greater than 5%. Seven more iterations were conducted and these iterations are shown in Table 1.

Table 1 Root of $f(x)=0$ as function of number of iterations for bisection method.

Iteration	x_ℓ	x_u	x_m	$\lvert\in_a\rvert\%$	$f(x_m)$
1	0.00000	0.11	0.055	----------	6.655×10^{-5}
2	0.055	0.11	0.0825	33.33	-1.622×10^{-4}
3	0.055	0.0825	0.06875	20.00	-5.563×10^{-5}
4	0.055	0.06875	0.06188	11.11	4.484×10^{-6}
5	0.06188	0.06875	0.06531	5.263	-2.593×10^{-5}
6	0.06188	0.06531	0.06359	2.702	-1.0804×10^{-5}
7	0.06188	0.06359	0.06273	1.370	-3.176×10^{-6}
8	0.06188	0.06273	0.0623	0.6897	6.497×10^{-7}
9	0.0623	0.06273	0.06252	0.3436	-1.265×10^{-6}
10	0.0623	0.06252	0.06241	0.1721	-3.0768×10^{-7}

At the end of 10th iteration, $\lvert\in_a\rvert = 0.1721\%$

Advantages and disadvantages of bisection method

Advantages

- The bisection method is always convergent. Since the method brackets the root, the method is guaranteed to converge.
- As iterations are conducted, the interval gets halved. So one can guarantee the error in the solution of the equation.

Disadvantages

- The convergence of the bisection method is slow as it is simply based on halving the interval.
- If one of the initial guesses is closer to the root, it will take larger number of iterations to reach the root.
- If a function $f(x)$ is such that it just touches the x-axis (Figure 6) such as

$$f(x) = x^2 = 0$$

it will be unable to find the lower guess, x_ℓ, and upper guess, x_u, such that

$$f(x_\ell)f(x_u) < 0$$

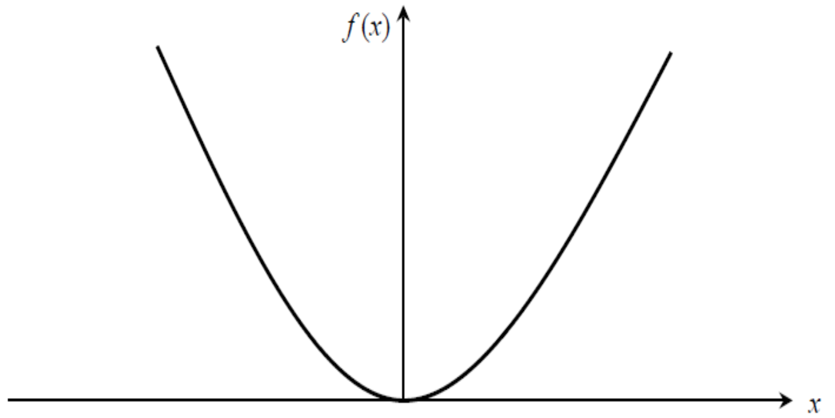

Figure 6 The equation $f(x) = x^2 = 0$ has a single root at $x = 0$ that cannot be bracketed.

References

- http://mathforcollege.com/nm/mws/gen/03nle/mws_gen_nle_txt_bisection.pdf
- Chapra, Steven C. Applied Numerical Methods with MATLAB for Engineers and Scientists. McGraw-Hill, 2017.
- Class Notes from ENGRD 3200: Engineering Computation taught by Professor Peter Diamessis at Cornell University