Bisection Method

The Bisection Method is a simple and reliable numerical technique used to find the solution to equations. In essence, it's like "guessing" where the answer might be, but each guess gets you closer to the correct result by cutting the range in half repeatedly until you get close enough.

How It Works

- Imagine you have a function, and you're trying to find the point where it crosses the x-axis (the point where the output value is zero). The Bisection Method works by taking two guesses, one on each side of the solution (meaning one where the function gives a positive value and one where it gives a negative value), and then it keeps cutting the range in half until the correct answer is found.

In computer graphics, especially in ray tracing, the Bisection Method can be used to find intersections between light rays and surfaces. If the equation defining a surface is too complex to solve analytically, the Bisection Method can be used to iteratively find where a ray hits the surface by narrowing down where the equation changes sign, helping with shadows, reflections, and lighting effects.

Example in Real Life: Finding a Broken Network Cable

- Let’s say you’re trying to find where a network cable is broken. The cable runs between two servers, and somewhere in the middle, there's a fault, but you don’t know exactly where. Here’s how the Bisection Method could help you:

Step 1 – Start with Two Guesses:

- - You know the cable works at Server A (the starting point), and you know it’s not working at Server B (the endpoint). These are like the two points that the Bisection Method starts with.

Step 2 – Test the Middle:

- - You pick a point halfway between the two servers and test the cable there.
- - If the cable works at this midpoint, you know the problem is in the second half of the cable, closer to Server B.
- - If the cable doesn’t work, you know the problem is in the first half, closer to Server A.

Step 3 – Narrow the Range:

- - You now repeat this process: testing the midpoint of the faulty half of the cable, narrowing down the location of the break each time by cutting the range in half.
- - Each step gets you closer to the exact location of the fault.

Step 4 – Find the Break:

- - After several steps, you will have a small enough range that you know exactly where the break is located. This is how the Bisection Method works—it narrows down the solution range by repeatedly cutting the interval in half.

The Bisection Method is like playing a game of hot-and-cold with math. You know the answer lies somewhere between two points, and by cutting the interval in half each time, you can quickly home in on the solution. Whether you're finding a broken cable or calculating light effects in a game, this simple but powerful technique helps you solve problems more efficiently.

SOLUTIONS OF EQUATIONS IN ONE VARIABLE

We consider one of the most basic problems of numerical approximation, the **root-finding problem**. This process involves finding a **root**, or solution, of an equation of the form $f(x) = 0$, for a given function f.
We can solve equations:

$x^3 + 4x^2 - 10 = 0$ on the interval [1, 2] ,

$x \cos x - 2x^2 + 3x - 1 = 0$ on the interval [1.2, 1.3] ,

$(x - 2)^2 - \ln x = 0$ on the intervals [1, 2] and [e, 4].

MATHEMATICAL AND GRAPHICAL REPRESENTATION

In computer science, the process of dividing a set continually in half to search for the solution to a problem, as the bisection method does, is known as a *binary search* procedure.

The first technique to solve equations like above, based on the Intermediate Value Theorem, is called the **Bisection**, or **Binary-search, method**.
Suppose $f(x)=0$ is a continuous function defined on $[a, b]$, with $f(a)$ and $f(b)$ of opposite sign. The Intermediate Value Theorem implies that a number p exists in (a, b) with $f(p) = 0$.

The bisection method calls for bisecting of subintervals of [*a*, *b*] and, at each step, locating the half containing *p*.

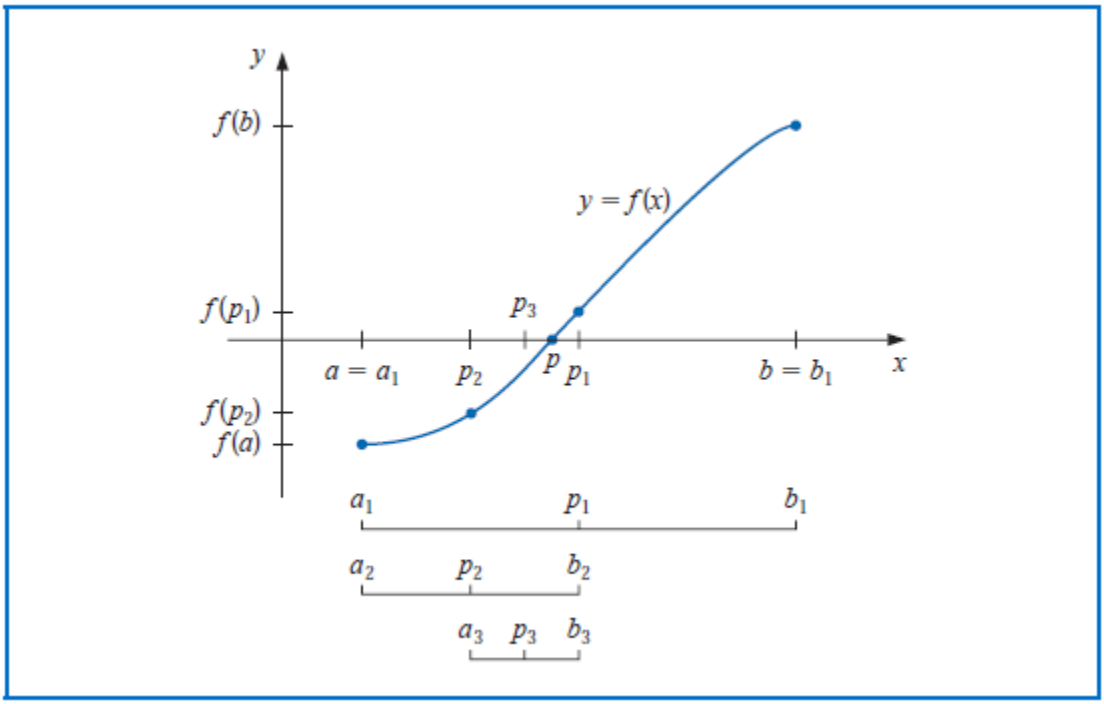
The first technique, based on the Intermediate Value Theorem, is called the **Bisection**, or **Binary-search, method**.

Suppose f is a continuous function defined on the interval $[a, b]$, with $f(a)$ and $f(b)$ of opposite sign. The Intermediate Value Theorem implies that a number p exists in (a, b) with $f(p) = 0$. 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. The method calls for a repeated halving (or bisecting) of subintervals of $[a, b]$ and, at each step, locating the half containing p.

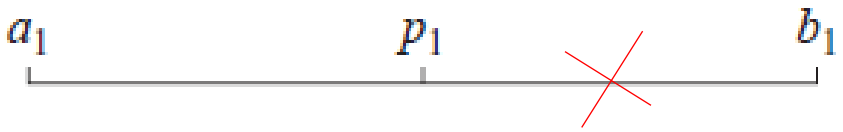
To begin, set $a_1 = a$ and $b_1 = b$, and let p_1 be the midpoint of $[a, b]$; that is,

$$p_1 = a_1 + \frac{b_1 - a_1}{2} = \frac{a_1 + b_1}{2}.$$

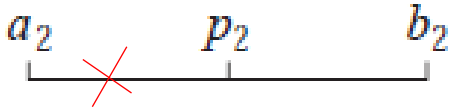
- If $f(p_1) = 0$, then $p = p_1$, and we are done.
- If $f(p_1) \neq 0$, then $f(p_1)$ has the same sign as either $f(a_1)$ or $f(b_1)$.
  - If $f(p_1)$ and $f(a_1)$ have the same sign, $p \in (p_1, b_1)$. Set $a_2 = p_1$ and $b_2 = b_1$.
  - If $f(p_1)$ and $f(a_1)$ have opposite signs, $p \in (a_1, p_1)$. Set $a_2 = a_1$ and $b_2 = p_1$.



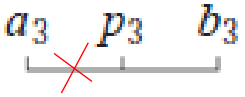
To begin, set $a_1 = a$ and $b_1 = b$, and let, $p_1 = \frac{a+b}{2}$



Let $f(a_1)$ and $f(p_1)$ have different signs but $f(p_1)$ and $f(b_1)$ have same signs, then the root lies in the interval $[a_1, p_1]$. Set $a_1 = a_2$ and $p_1 = b_2$, and let $p_2 = \frac{a_2+b_2}{2}$



If $f(a_2)$ and $f(p_2)$ have same signs but $f(p_2)$ and $f(b_2)$ have different signs, then the root lies in the interval $[p_2, b_2]$. Then find $p_3 = \frac{a_3+b_3}{2}$ on the same lines



Determining the successive values of $p_4, p_5, p_6, \ldots$ there exists p_n for some *n* such that $f(p_n)$ is very close to 0. Such p_n will be required root of the equation $f(x) = 0$.

Example 1

If we have an equation in the form $cos(t) = 3t - 1$, put it equal to 0, that is

$$f(t) = \cos(t) - 3t + 1 = 0.$$

Find the interval [a, b] as [0, 1] of solution for $f(t) = \cos(t) - 3t + 1$ with $f(0) = 2 > 0$ and $f(1) = -1.46 < 0$, then use

$$p_i = \frac{a+b}{2},\ i = 0,1,2, \ldots$$

First Iteration p_1 : First find

$$p_1 = \frac{a+b}{2} = \frac{0+1}{2} = 0.5$$

Now we will substitute $p_1 = 0.5$, in $f(t)$, and we have

$$f(0.5) = \cos(0.5) - 3(0.5) + 1 = 0.3776 > 0$$

As we know that $f(0) > 0$, like $f(0.5)$ so we will replace 0.5 by 0 in the interval. We have the new interval as [0.5, 1] and it completes First Iteration.

Second Iteration $i = 2$:

$$p_2 = \frac{a+b}{2} = \frac{0.5+1}{2} = 0.75$$

We will substitute $p_2 = 0.75$, in $f(t)$, implies

$$f(0.75) = \cos(0.75) - 3(0.75) + 1 = -0.5183 < 0$$

As we know at $t = 1$, $f(t) < 0$, so we will replace 0.75 by 1 in the interval and we have new interval [0.5,0.75]. It completes second Iteration.

In this way we will proceed further to find $p_3, p_4, p_5, \ldots$

until the gap between two p_i is very less (say within 10^{-3})

$$|b - a| < 10^{-3} (= 0.001)$$

or any other given criteria like

$$\text{(i) } |p_i - p_{i-1}| < \varepsilon \quad \text{or} \quad \text{(ii) } |f(p_n)| < \varepsilon,$$

As we get $p_3, p_4, p_5, p_6, \ldots p_{12} = 0.625, 0.5625, 0.59375, 0.6094, \ldots, 0.6072$ with $f(p_{12}) = -0.00035$ with required accuracy as the solution.

Example 2

Use bisection method to find solution accurate to within 10^{-2} for

$x^3 - 7x^2 + 14x - 6 = 0$ on the interval [0, 1] .

Solution

Let us mark $a = 0$ and b = 1, and for $f(x) = x^3 - 7x^2 + 14x - 6$, we know that

$$f(0) = -6 < 0 \text{ and } f(1) = 2 > 0.$$

So for the first iteration: $p_1 = \frac{a+b}{2} = \frac{0+1}{2} = 0.5$, with

$$f(0.5) = -0.625 < 0$$

As at $x = 0$, $f(x) < 0$, so we will replace 0 with 0.5 in the interval. So we have new interval [0.5, 1] completing the First Iteration.

Second Iteration: $p_2 = \frac{a+b}{2} = \frac{0.5+1}{2} = 0.75$

Now we will substitute $p_2 = 0.75$, in $F(x)$, we have

$$f(0.75) = 0.9844 > 0$$

As we know at $x = 1$, $f(x) > 0$, so we will put 0.75 in place of 1 in the interval.

We can summarize these iterations in the form of a table as:

n	a_n	b_n	p_n	$f(p_n)$
1	0	1	0.5	-0.625
2	0.5	1	0.75	0.9844
3	0.5	0.75	0.625	0.2598
4	0.5	0.625	0.5625	-0.1619
5	0.5625	0.625	0.5938	0.0544
6	0.5625	0.5938	0.5782	-0.0521
7	0.5782	0.5938	0.5860	0.0015
8	0.5782	0.5860	0.5821	-0.025
9	0.5821	0.5860	0.5841	-0.0115

Since, $ERROR = |p_9 - p_8| = 0.002$, So the approximated solution is $x = 0.58$.

PROBLEM STATEMENT

Imagine you are working as a software developer tasked with optimizing a computer network. The system's performance is heavily dependent on balancing network traffic to avoid bottlenecks. To model the traffic, a nonlinear equation such as $x - 2^{-x} = 0$ can represent how traffic flow decreases exponentially based on certain conditions (like increasing congestion). You need to find the exact balance point of traffic, where the system stabilizes. Using the Bisection Method, you can accurately calculate this balance point within the interval [0, 1], helping ensure that your network operates efficiently and without congestion. Let's find the solution to this equation, accurate to within 10^{-5}, so you can apply it to your traffic management system.

Solution

Let us mark $a = 0$ and b $= 1$, and for

$$f(x) = x - 2^{-x}, \qquad f(0) = -1 < 0 \qquad \text{and} \qquad f(1) = 0.5 > 0.$$

n	a_n	b_n	p_n	$f(p_n)$
1	0	1	0.5	-0.2071
2	0.5	1	0.75	0.1554
3	0.5	0.75	0.625	-0.0234
4	0.625	0.75	0.6875	0.0666
5	0.625	0.6875	0.65625	0.02172
6	0.625	0.65625	0.640625	-0.00081
7	0.640625	0.65625	0.64844	0.0105
8	0.640625	0.64844	0.64453	0.005
9	0.640625	0.64453	0.64258	0.002

10	0.640625	0.64258	0.64160	0.0006
11	0.640625	0.64160	0.64111	-0.00011
12	0.64111	0.64160	0.641355	0.000245
13	0.64111	0.641355	0.64123	0.00006
14	0.64111	0.64123	0.64117	-0.000023
15	0.64117	0.64123	0.64120	0.00002

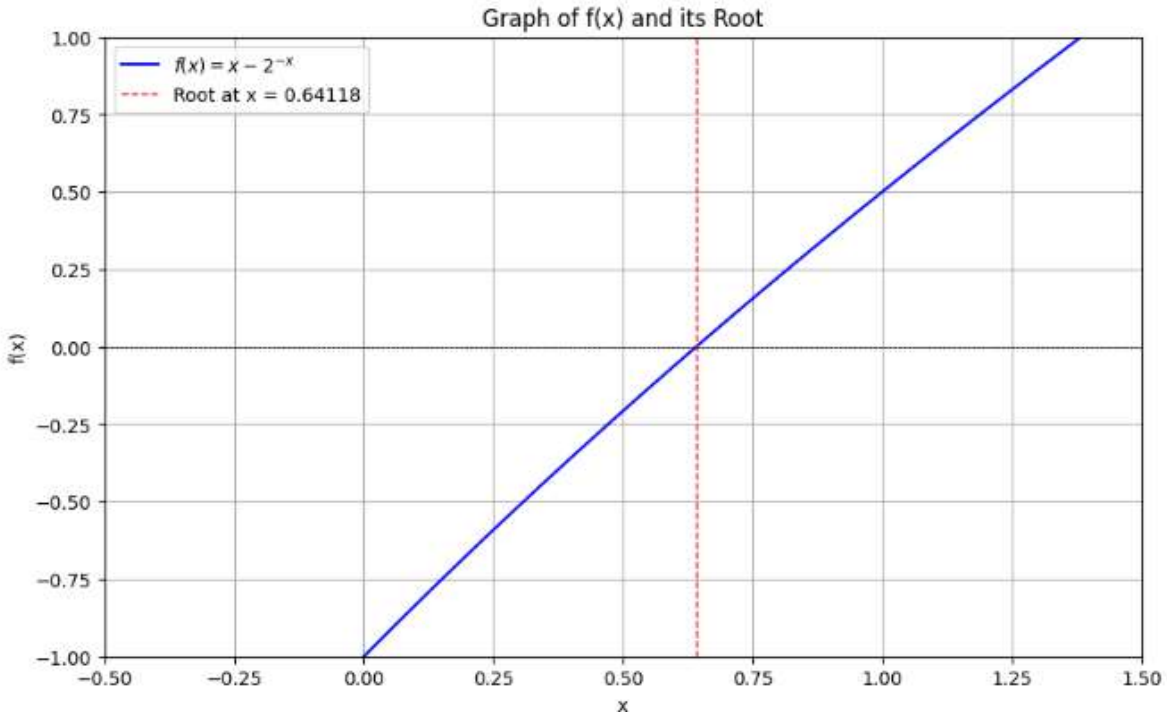
So the approximated solution is $x = 0.64120$.

Python Code

```python
# Function representing the equation f(x) = x - 2^(-x)
def f(x):
    return x - 2**(-x)

# Bisection Method implementation
def bisection_method(a, b, tol):
    if f(a) * f(b) > 0:
        print("Bisection method cannot proceed. The function must have opposite signs at a and b.")
        return None

    while (b - a) / 2 > tol:
        midpoint = (a + b) / 2
        print(f"Current midpoint: {midpoint}, f(midpoint): {f(midpoint)}")

        if f(midpoint) == 0:
            return midpoint  # We've found the exact root
        elif f(a) * f(midpoint) < 0:
            b = midpoint  # The root lies in the left half
        else:
            a = midpoint  # The root lies in the right half

    return (a + b) / 2  # Return the approximate root

# Define the interval and tolerance
a = 0
b = 1
tolerance = 1e-5

# Call the bisection method to find the root
solution = bisection_method(a, b, tolerance)

if solution is not None:
    print(f"Approximate solution: {solution:.5f}")
```

Results

```
Current midpoint: 0.5, f(midpoint): -0.20710678118654757
Current midpoint: 0.75, f(midpoint): 0.1553964424986395
Current midpoint: 0.625, f(midpoint): -0.02341977732550482
Current midpoint: 0.6875, f(midpoint): 0.066571093963258
Current midpoint: 0.65625, f(midpoint): 0.02172452140413339
Current midpoint: 0.640625, f(midpoint): -0.0008100080393891318
Current midpoint: 0.6484375, f(midpoint): 0.010466610801804
Current midpoint: 0.64453125, f(midpoint): 0.004830646247165338
Current midpoint: 0.642578125, f(midpoint): 0.0020109061142677964
Current midpoint: 0.6416015625, f(midpoint): 0.0006005958893881003
Current midpoint: 0.64111328125, f(midpoint): -0.00010466934958675012
Current midpoint: 0.641357421875, f(midpoint): 0.00024797244970042875
Current midpoint: 0.6412353515625, f(midpoint): 7.165384520102513e-05
Current midpoint: 0.64117431640625, f(midpoint): -1.6507178382529908e-05
Current midpoint: 0.641204833984375, f(midpoint): 2.7573476858777646e-05
Current midpoint: 0.6411895751953125, f(midpoint): 5.533185100881077e-06
Approximate solution: 0.64118
```



Disadvantage	Advantage
The Bisection method, though conceptually clear, has significant drawbacks. It is relatively slow to converge (that is, N may become quite large before $\lvert p - p_N \rvert$ is sufficiently small), and a good intermediate approximation might be inadvertently discarded.	The method has the important property that it always converges to a solution.

PRACTICE

Use the Bisection method to find solutions accurate to within 10^{-2} for $x^4 - 2x^3 - 4x^2 + 4x + 4 = 0$ on each interval.

a. $[-2, -1]$ **b.** $[0, 2]$ **c.** $[2, 3]$ **d.** $[-1, 0]$

Use the Bisection method to find solutions accurate to within 10^{-5} for the following problems.

a. $x - 2^{-x} = 0$ for $0 \le x \le 1$

b. $e^x - x^2 + 3x - 2 = 0$ for $0 \le x \le 1$

c. $2x\cos(2x) - (x+1)^2 = 0$ for $-3 \le x \le -2$ and $-1 \le x \le 0$

d. $x\cos x - 2x^2 + 3x - 1 = 0$ for $0.2 \le x \le 0.3$ and $1.2 \le x \le 1.3$