Math 311

Numerical Methods

2.1: The Bisection Method

Solutions of Equations in One Variable

S. K. Hyde

Burden and Faires, any ed.

Winter 2025

1 Introduction

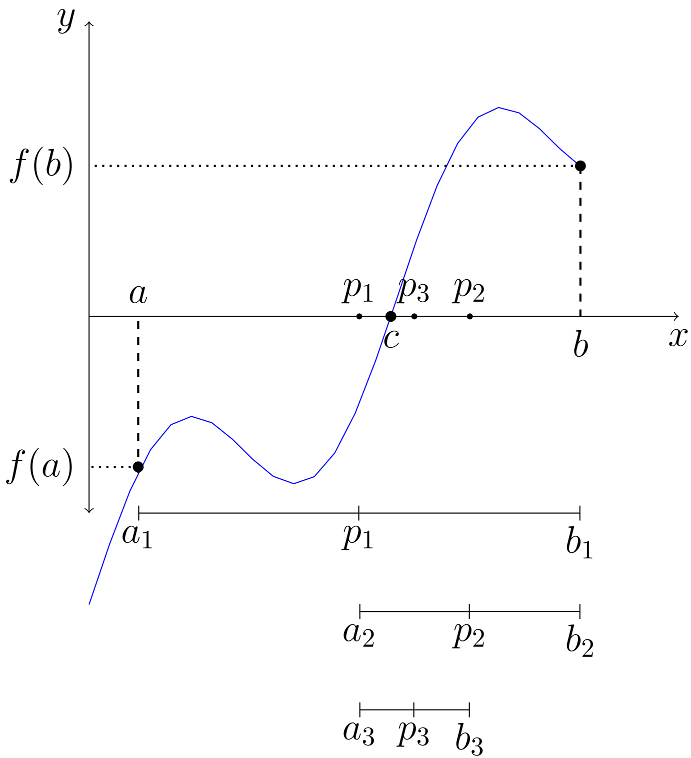
- A basic way of solving an equation of one variable $f(x) = 0$.
- The solution is called the root x of the equation $f(x) = 0$.
- It is also called a zero of $f(x)$
- The first method we will learn is the Bisection Method (or Binary-search method).
- Motivated by the Intermediate Value theorem.
- Example of this: The High-Low game.

The High Low Game

Let's play a game of "High-Low". I will try to guess your number in 7 tries. How? Follow these steps:

1. The number is between 1 and 100, so I guess the midpoint. That's ≈ 50. So I would guess 50 or 51 (depending on if I decide to round or truncate) and then ask if the guess is <u>Too High</u> or <u>Too Low</u>. Suppose I chose 50.
   - If it was too high, then your number is in [1,50]. My next guess $\frac{1+50}{2} \approx 26$
   - If it was too low, then your number is in [50,100]. I guess $\frac{50+100}{2} = 75$
2. Now, you repeat step, keeping the real answer bracketed in an interval. On each step, you guess the midpoint of the current interval.
3. For a number between 1 and 100, this will guarantee a correct answer within 7 tries. Try it out yourself!

1.1 How does Bisection work?



Bisection Algorithm

- Input: a, b, TOL, max number of iterations N_0
- Output: Approximate solution p or message of failure

Step 1 for (i = 1 to N_0) do steps 2-4

  Step 2 set $p = \frac{a+b}{2}$

  Step 3 if ($f(p) = 0$ or $(b - a)/2 < TOL$, then output(p); stop;

  Step 4 if $f(a) \cdot f(p) > 0$, then set $a = p$ else set $b = p$

Step 5 output("Method failed after N_0 iterations"); stop;

```
##########################################################################
# Bisection Method
#   Implements the bisection method for finding a root
##########################################################################

bisection = function(a,b,f,eps=1e-7,n=30) {
    ##
    ## Inputs
    ##   [a,b] = interval root is bracketed by
    ##   f = function to find the root of
    ##   eps = tolerance (defaults to 1e-9 = .000000001)
    ##   n = iterations allowed (defaults to 30)

    if ( sign(f(a))*sign(f(b)) > 0 )
        stop(paste("root does not exist in [",a,",",b,"]",sep=""))

    fail=TRUE
    save=c("Midpt (p)","LeftB (a)","RightB (b)","err (b-a)")
    piter = matrix(0,0,length(save))  #initialize a matrix to save iterations
    for (i in 1:n) {
        p = (a+b)/2
        piter = rbind(piter,c(p,a,b,abs(b-a)/2)) #save stuff
        if ( f(p)==0 || (b-a)/2 < eps ) { fail=FALSE; break }
        if ( sign(f(a))*sign(f(p)) > 0 ) { a=p } else { b=p }
    }
    if (fail) warning(paste("Failed to converge after",i,"iterations"))
    dimnames(piter)=list(1:i,save)
    return(list(iterations=piter,zero=p))
}
```

Convergence of Bisection Method

Theorem. *2.1. If* $f \in C[a, b]$ *and suppose* $f(a) \cdot f(b) < 0$*. The Bisection Algorithm generates a sequence* $\{p_n\}$ *approximating* p *with the property*

$$|p_n - p| \leqslant \frac{b-a}{2^n}, \text{ for } \ n \geqslant 1$$

Proof. This is what the n^{th} interval looks like:

The algorithm halves the interval on each stgep

- Start our sequence at $n = 1$. At the beginning, the length of the interval is $b - a$.
- When $n = 2$, the length is $\frac{b-a}{2}$. So, after n times, we have

$$b_n - a_n = \frac{b-a}{2^{n-1}}, \text{ where } p \in (a_n, b_n)$$

- Since $p_n = \dfrac{a_n + b_n}{2}$, then $|p_n - p| \leqslant \dfrac{1}{2}(b_n - a_n) = \dfrac{1}{2}\left(\dfrac{b-a}{2^{n-1}}\right) = \dfrac{b-a}{2^n}$.
- Thus, as $n \to \infty$, $p_n \to p$ at the rate of $\mathcal{O}(2^{-n})$

$\square$

1.2 Example

How many iteration of Bisection would be required to get the approximation accurate to within 10^{-8}. Suppose $a = 0$ and $b = 1$.

Proof. Using Theorem 2.1, we see that

$$|p_n - p| \leqslant \frac{b-a}{2^n} < 10^{-8} \Longrightarrow 2^{-n} < 10^{-8}$$

$$-n \log 2 < -8 \log 10$$

$$n > \frac{8}{\log 2} = 26.57 \approx 27 \text{ iterations.}$$

□

- The BEST thing about Bisection method is that it GUARANTEES convergence.
- That is a quality that is very desirable, but is rare for numerical methods.
- Problem is that it is SLOW. There are faster methods.
- Another problem. Functions that barely touch the x axis but never cross are not possible to solve using Bisection method.
- For example, $f(x) = x^3 - 3x + 2$. Check a graph out on it. It is impossible to find the zero at $x = 1$ using Bisection.
- We are now going to talk more about Fixed point methods.