

Lecture 04 - A

§3.1 Nonlinear Problems and the Bisection Method

T. Takahashi

Department of Mathematics and Statistics
University of New Mexico



We will answer these questions:

- What makes solving nonlinear problems difficult?
- What are some rudimentary strategies to find general locations of roots?
- How does the bisection method find a root?
- How can we describe how quickly it finds the root?



Rootfinding



Linear vs. Nonlinear

- We can solve a **Linear problem** $Ax = b$, e.g.

$$\begin{cases} 2x + 2y = 4 \\ -3x + 5y = 2 \end{cases} \quad (1)$$

- How about this **nonlinear problem**?

$$\begin{cases} x^2 + y = 4 \\ \cos(x) + y^2 = 2 \end{cases} \quad (2)$$

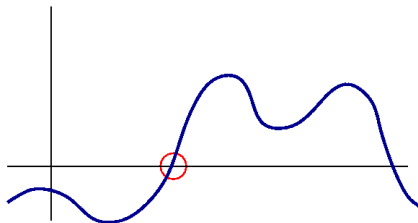
Question

- ▶ Can you solve the nonlinear problem by hand?
- ▶ How many solutions are there?
- ▶ Is there even a solution?
- ▶ How can we systematically approach the problem?

Root Finding

Definition: Root finding

Given a function $f(x)$, the **root finding problem** seeks the number(s) x such that $f(x) = 0$. The number x is called the **root**.



Difficulty when $f(x)$ is nonlinear:

- An analytic solution may not exist
- There may be multiple solutions
- There may be no solutions at all



Strategy: Iterate

- For most nonlinear equations an algebraic “one-shot” solution **does not exist**¹. Consider solving the equation:

$$\cos(x) = x$$

- Only option is to approximate the root.

Strategy

Start with an initial guess x_0 . Sequentially improve

$$x_0, x_1, x_2, \dots$$

in a way that makes $f(x_k)$ smaller and smaller. If $f(x_k) \approx 0$, we are done.



¹Few exceptions: quadratics, cubics, etc.

Root Finding: Solving nonlinear equations

Goals:

- Systematically approximate roots to 1D functions
- Compare convergence properties of different methods
- Understand the pros and cons of each method

Root-finding methods:

- 1 Bisection Method
- 2 Newton's Method
- 3 Secant Method



Bracketing



Question

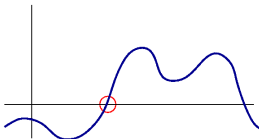
Nonlinear solvers require an initial guess x_0 . What can we do get a decent starting point?



Question

Nonlinear solvers require an initial guess x_0 . What can we do get a decent starting point?

- Just plot the function



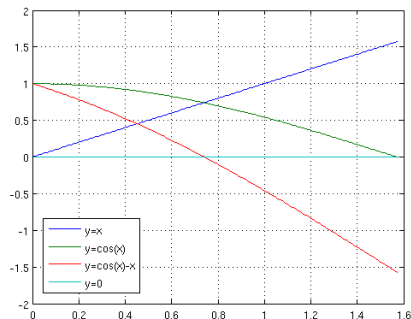
- Look for the location where $f(x)$ crosses the x -axis
- Works well for one-dimensional problem

Plotting Example

Example

Obtain an approximate solution $\cos(x) = x$ by plotting.

- **Option 1:** Plot $y_1 = x$ and $y_2 = \cos x$ and see where they intersect
- **Option 2:** Plot $y = \cos x - x$ and see where it crosses the x-axis



- Decent guess $x \approx 0.75$



Intermediate Value Theorem

Intermediate Value Theorem

Suppose $f(x)$ is continuous on $[a, b]$. If

$$f(a) < s < f(b)$$

then there exists at least one point $c \in (a, b)$ such that

$$f(c) = s.$$

Let $s = 0$

Suppose $f(x)$ is continuous on $[a, b]$. If

$$f(a) < 0 < f(b) \implies (f \text{ changes from negative to positive})$$

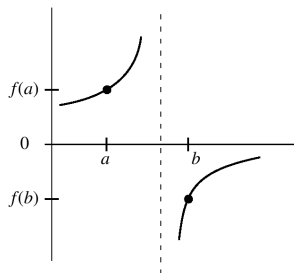
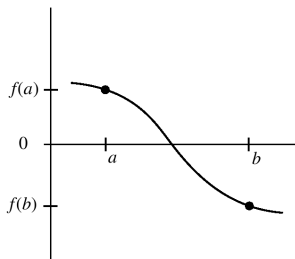
then there exists at least one point $c \in (a, b)$ such that

$$f(c) = 0 \implies (f \text{ root lives somewhere inside } (a, b))$$

Bracket

Definition: Bracket

- A root x is **bracketed** on $[a, b]$ if $f(a)$ and $f(b)$ have opposite sign
- We call $[a, b]$ a **bracket** for the root.
- Bracketing only works when $f(x)$ is continuous



Key Idea

If f changes signs from a to b , then a root must exist in between.

Bracket Algorithm

Steps:

- 1 Subdivide the interval into n chunks.
- 2 Check each subinterval for a sign change.

```
1 % Given: f(x), x_min, x_max, n
2 dx = (x_max - x_min) / n;
3 x_left = x_min;
4 i = 0;
5
6 while (i < n)
7     i = i + 1;
8     x_right = x_left + dx;
9     if (f(x) changes sign on [x_left, x_right])
10         save [x_left, x_right] as an interval with a root;
11     end
12     x_left = x_right;
13 end
```



How to Check for a Bracket

Option 1

```
fa = myfunc(a);  
fb = myfunc(b);  
if (fa * fb < 0)  
    % Save the bracket [a, b]  
end
```

Option 2

```
if (sign(fa) ~= sign(fb))  
    % Save the bracket [a, b]  
end
```

Why option 2 is better:

- **Efficiency:** The function `sign()` only considers the sign bit
- **Safety:** Avoids potential underflow for small $f(a) * f(b)$



Method: Bracketing

- **Given:** $f(x)$, interval $[x_{\min}, x_{\max}]$, number of subintervals n
- **Output:** List of intervals (brackets) containing roots.
- **Idea:** Find general locations of the root without plotting.

Key Idea

- For rootfinding, we need a starting point
- Plotting is a qualitative approach for preliminary investigation
- Bracketing is quantitative and also works for higher dimensions



Bisection Method



High or Low

High or Low Game

- **Goal:** Guess my number.
- **Rule:** The number is an integer between 1 and 100. You guess a number, and I tell you whether your guess is too low or too high.



High or Low

High or Low Game

- **Goal:** Guess my number.
- **Rule:** The number is an integer between 1 and 100. You guess a number, and I tell you whether your guess is too low or too high.

Optimal play:

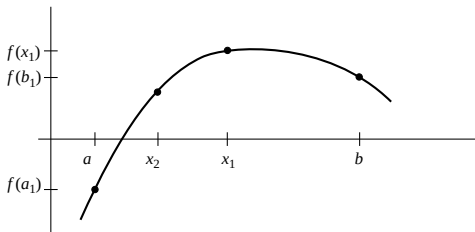
- Start with 50.
- If the guess is too high, next guess should be 25.
- If the guess is too low, next guess should be 75.
- Continue by halving the search interval.

Key Idea

You are guaranteed to eliminate half of the wrong answers.



Bisection Idea



Play the **high or low** game to find the root of $f(x)$:

- Starting with bracket $[a, b]$.
- Cut the search interval in half. Is the root to the left or right?
- Choose the half interval containing the root.
- Repeat.



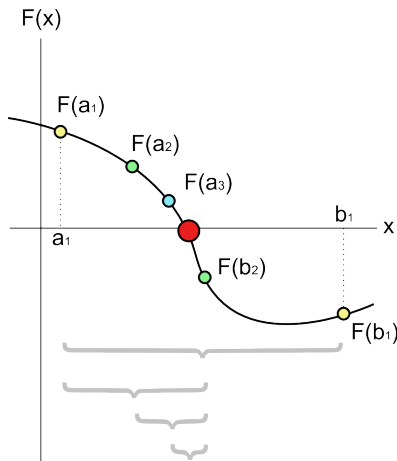
Midpoints and Sign Test

The **midpoint** of bracket $[a, b]$ is

$$x_m = \frac{a + b}{2}$$

Main steps:

- 1 Compute the midpoint x_m .
- 2 Evaluate $f(x_m)$ and compare with $f(a)$.
 - ▶ If $\text{sign}(f(x_m)) = \text{sign}(f(a))$, set $a = x_m$.
 - ▶ Otherwise, set $b = x_m$.
- 3 Repeat until convergence.



Key Idea

Always maintain the bracket.

Method: Bisection

- **Given:** $f(x)$, bracket $[x_a, x_b]$, tolerance ϵ
- **Output:** Approximate solution to $f(x) = 0$.
- **Idea:** Reduce search interval by half each iteration.

Why we like bracketing:

- Easy to implement
- Predictable performance
- A variant of this method can be extended to higher dimensions



Bisection Update Procedure

```
1 % user provides a and b such that sign(f(a)) != sign(f(b))
2 for k = 1, 2, ..., until convergence
3     x_m = (a+b)/2;           % Compute the midpoint
4     if (sign(f(x_m))==sign(f(a)))
5         a = x_m;             % Root is in [x_m, b]
6     else
7         b = x_m;             % Root is in [a, x_m]
8     end
9 end
```

- Use `sign` to compare the sign of $f(x_m)$ and $f(a)$
- If they are the same, let x_m be the new left end
- Otherwise, let x_m be the the right end



Bisection Numerical Result

Solve with bisection

$$f(x) = x - x^{1/3} - 2 = 0$$

Initial bracket: $f(a) = f(3) \approx -0.4422$ and $f(b) = f(4) \approx 0.4126$

k	a	b	x_{mid}	$f(x_{mid})$
0	3	4		
1	3	4	3.5	-0.01829449
2	3.5	4	3.75	0.19638375
3	3.5	3.75	3.625	0.08884159
4	3.5	3.625	3.5625	0.03522131
5	3.5	3.5625	3.53125	0.00845016
6	3.5	3.53125	3.515625	-0.00492550
7	3.51625	3.53125	3.5234375	0.00176150
8	3.51625	3.5234375	3.51953125	-0.00158221
9	3.51953125	3.5234375	3.52148438	0.00008959
10	3.51953125	3.52148438	3.52050781	-0.00074632



Bisection by Hand

Example

Approximate the root of $f(x) = x^2 - 3x + 1$ using the starting bracket $[1, 5]$ with two iterations of the bisection method.

Start with $a = 1$, and $b = 5$. $f(a) = -1$ and $f(b) = 11$

- 1 Find the midpoint: $m = \frac{a+b}{2} = 3$.
- 2 Check the sign: $f(m) = f(3) = 1 > 0$
- 3 Choose the replacement: since $f(b) = 11 > 0$, let $b = 3$

$$a = 1, \quad b = 3$$

- 4 Find the midpoint: $m = \frac{a+b}{2} = 2$.
- 5 Check the sign: $f(2) = -1 < 0$
- 6 Choose the replacement: since $f(a) = -1 < 0$, let $a = 2$

$$a = 2, \quad b = 3$$

(Note: the answer is approximately 2.61803)



Error Analysis



Analysis of Bisection

- Let δ_n be the length of the n^{th} bracket:

$$\delta_0 = b - a \quad (\text{initial bracketing interval})$$

$$\delta_1 = \frac{1}{2} \delta_0$$

$$\delta_2 = \frac{1}{2} \delta_1 = \frac{1}{4} \delta_0$$

$$\vdots$$

$$\delta_n = \left(\frac{1}{2}\right)^n \delta_0$$

- Thus, the relative reduction in error is:

$$\frac{\delta_n}{\delta_0} = \left(\frac{1}{2}\right)^n = 2^{-n}$$

- or equivalently,

$$n = -\log_2 \left(\frac{\delta_n}{\delta_0} \right).$$



Interpretation of Results

$$\frac{\delta_n}{\delta_0} = \left(\frac{1}{2}\right)^n$$

The ratio of the n^{th} interval to the original interval is the n^{th} power of 0.5.

$$n = -\log_2 \left(\frac{\delta_n}{\delta_0} \right)$$

For some desired ratio δ_n/δ_0 , it requires $|\log_2(\text{ratio})|$ steps to achieve that reduction.

	n	δ_n/δ_0
Example:	5	3.1×10^{-2}
	10	9.8×10^{-4}
	20	9.5×10^{-7}
	30	9.3×10^{-10}
	40	9.1×10^{-13}
	50	8.9×10^{-16}



Bisection Error

- At iteration n the true answer must be in the range

$$a_n < x^* < b_n$$

so the error estimate is

$$e_n = |x_* - c_n| \leq \frac{b_n - a_n}{2} = \frac{\delta_n}{2}.$$

- Recall that

$$\delta_n = \left(\frac{1}{2}\right)^n \delta_0.$$

- Thus, the error after n steps is bounded by

$$|x_* - c_n| \leq \left(\frac{1}{2}\right)^{n+1} \delta_0 = \left(\frac{1}{2}\right)^{n+1} (b - a).$$



Bisection Error Theorem

Theorem

If the bisection algorithm is applied to $f \in C([a, b])$ where $f(a)f(b) < 0$, then after n steps the error estimate is

$$e_n < \frac{b-a}{2^{n+1}}.$$

- No need to consider backward error
- We have an error bound for the forward error
- The user may provide an error tolerance ϵ
 - Run bisection until $e_n \leq \epsilon$



Bisection: Example

Example

What is least amount of bisection iterations needed to guarantee that the error is less than $\epsilon = 10^{-8}$ if the initial bracket is $a = -2$ and $b = 3$?

Begin by setting error e_n less than tolerance ϵ

$$\frac{b-a}{2^{n+1}} < 10^{-8}$$

$$\Rightarrow \frac{5}{2^{n+1}} < 10^{-8} \quad \text{substitute } a, b$$

$$\Rightarrow 2^{-(n+1)} < 10^{-8}/5 \quad \text{divide by 5}$$

$$\Rightarrow -(n+1) < \log_2(10^{-8}/5) \quad \text{log base 2 both sides}$$

$$\Rightarrow n > -\log_2(10^{-8}/5) - 1 \quad \text{note the inequality flip}$$

$$\Rightarrow n > 27.89$$

So, want at least $n = 28$ steps for a guarantee.



Convergence Rate



Convergence Rate

Definition: Convergence Rate

Let $e_n = |x^* - x_n|$ be the error of an iterative method. The method is said to converge with rate r if

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^r} \leq C$$

Special Cases:

- If $r = 1$ and $C < 1$, then the rate is linear
- If $r = 2$ and $C > 0$, then the rate is quadratic
- If $r = 3$ and $C > 0$, then the rate is cubic



Determine Convergence Rate

- Rule of thumb: convergence rate is r if

$$\frac{(\text{new error})}{(\text{old error})^r} \approx C$$

Linear

If $\frac{\text{new error}}{\text{old error}} \approx C$, then convergence is linear

Quadratic

If $\frac{(\text{new error})}{(\text{old error})^2} \approx C$, then convergence is quadratic



Example

Example

Find the convergence rate for each of the following sequences

- 1 $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} \dots$
- 2 $10^{-2}, 10^{-4}, 10^{-6}, 10^{-8} \dots$
- 3 $10^{-2}, 10^{-4}, 10^{-8}, 10^{-16} \dots$
- 4 $10^{-2}, 10^{-6}, 10^{-18}, \dots$



Example

Example

Find the convergence rate for each of the following sequences

- ① $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} \dots$ (linear with $C = 10^{-1}$)
- ② $10^{-2}, 10^{-4}, 10^{-6}, 10^{-8} \dots$ (linear with $C = 10^{-2}$)
- ③ $10^{-2}, 10^{-4}, 10^{-8}, 10^{-16} \dots$ (quadratic)
- ④ $10^{-2}, 10^{-6}, 10^{-18}, \dots$ (cubic)



Example

Example

Find the convergence rate for each of the following sequences

- 1 $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} \dots$ (linear with $C = 10^{-1}$)
- 2 $10^{-2}, 10^{-4}, 10^{-6}, 10^{-8} \dots$ (linear with $C = 10^{-2}$)
- 3 $10^{-2}, 10^{-4}, 10^{-8}, 10^{-16} \dots$ (quadratic)
- 4 $10^{-2}, 10^{-6}, 10^{-18}, \dots$ (cubic)

How fast is bisection converging?



Convergence rate of Bisection Algorithm

- We know the error estimate

$$|e_n| = |x_* - c_n| \leq \frac{\delta_0}{2^{n+1}}$$

- Thus the ratio of new to old error is

$$\frac{|e_{n+1}|}{|e_n|} \approx \frac{\frac{\delta_0}{2^{n+2}}}{\frac{\delta_0}{2^{n+1}}} = \frac{1}{2}$$



Convergence rate of Bisection Algorithm

- We know the error estimate

$$|e_n| = |x_* - c_n| \leq \frac{\delta_0}{2^{n+1}}$$

- Thus the ratio of new to old error is

$$\frac{|e_{n+1}|}{|e_n|} \approx \frac{\frac{\delta_0}{2^{n+2}}}{\frac{\delta_0}{2^{n+1}}} = \frac{1}{2}$$

Convergence Rate

Bisection method converges linearly with rate $C = 1/2$.



Questions

- 1 What about bisection method makes it a good root-finding algorithm?
- 2 Does this algorithm ever fail?
- 3 Is there a disadvantage to bisection method?
- 4 Is it a fast algorithm?

