Roots of Equations

- Real roots of single variable function.
- Closed domain methods (bracketing).

Examples of nonlinear equations

- Equations of a single variable:

$$x^2 - 6x + 9 = 0$$
$$x^2 - \cos(x) = 0$$
$$\exp(x)\ln(x^2) - x\cos(x) = 0$$

- Equations of two variables:

$$y(x^3 - 1) = x^4$$
$$x^2 + y^2 = 1$$

Introduction

Given a continuous non linear function $f(x)$, find the value $x = c$ such that $f(c) = 0$.

Given a continuous non linear function $f(x)$, find the value $x = c$ such that $f(c) = 0$.

The non-linear equation $f(x) = 0$ may be a:

- An algebraic equation (roots of polynomials).

Introduction

Given a continuous non linear function $f(x)$, find the value $x = c$ such that $f(c) = 0$.

The non-linear equation $f(x) = 0$ may be a:

- An algebraic equation (roots of polynomials).
- A transcendental equation.

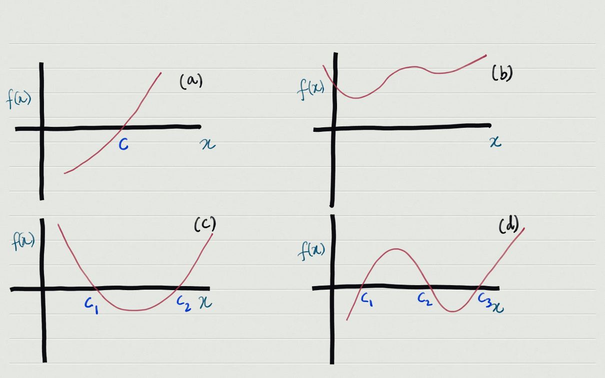
Introduction

Given a continuous non linear function $f(x)$, find the value $x = c$ such that $f(c) = 0$.

The non-linear equation $f(x) = 0$ may be a:

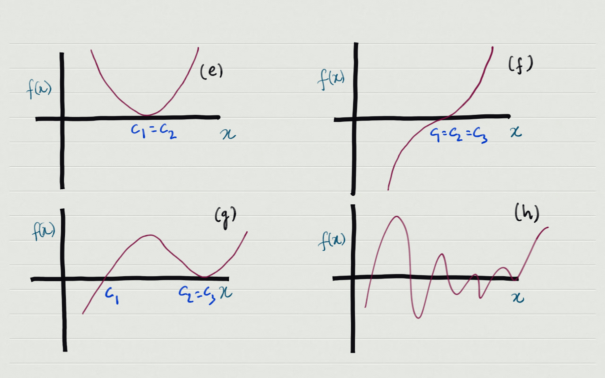
- An algebraic equation (roots of polynomials).
- A transcendental equation.
- ...

Behaviour of non-linear functions



(a) A single root.
(b) No real roots exist (complex roots might).
(c) Two simple roots.
(d) Three simple roots.

Behaviour of non-linear functions



(e) Two multiple roots.
(f) Three multiple roots.
(g) One simple root and two multiple roots.
(h) Multiple roots.

Some prelimenaries

- All non-linear equations have to be solved iteratively!

Some prelimenaries

- All non-linear equations have to be solved iteratively!
- We always start out with a guess – an approximate root.

Some prelimenaries

- All non-linear equations have to be solved iteratively!
- We always start out with a guess – an approximate root.
- Starting from the guess we iterate – the better the guess, the better our chances of converging to the solution and fewer the number of iterations required to converge to the solution.

Some prelimenaries

- All non-linear equations have to be solved iteratively!
- We always start out with a guess – an approximate root.
- Starting from the guess we iterate – the better the guess, the better our chances of converging to the solution and fewer the number of iterations required to converge to the solution.
- So, it is crucial to have a good guess...

Bounding the solution

- Bounding the solution involves finding a rough estimate of the solution that can be used as in initial guess in an iterative process that refines the solution to a specified tolerance.

Bounding the solution

- Bounding the solution involves finding a rough estimate of the solution that can be used as in initial guess in an iterative process that refines the solution to a specified tolerance.
- If possible, the root should be bracketed between two points at which the value of the non-linear function changes sign!

Bounding the solution

- Getting a good guess – seems like a chicken and egg problem – but not so.

Bounding the solution

- Getting a good guess – seems like a chicken and egg problem – but not so.
- GRAPH the functions to get an idea!! This is the best way to get an idea of the solution and get an approximate solution. In case of multiple solutions it also ensures that you have the solution that you want!

Bounding the solution

- Getting a good guess – seems like a chicken and egg problem – but not so.
- GRAPH the functions to get an idea!! This is the best way to get an idea of the solution and get an approximate solution. In case of multiple solutions it also ensures that you have the solution that you want! *People who omit the sketch throw away the most important problem solving tool they possess – geometric intuition.*

Bounding the solution

- Getting a good guess – seems like a chicken and egg problem – but not so.
- GRAPH the functions to get an idea!! This is the best way to get an idea of the solution and get an approximate solution. In case of multiple solutions it also ensures that you have the solution that you want! *People who omit the sketch throw away the most important problem solving tool they possess – geometric intuition.*
- Incremental search – ie start with different guesses and search.

Bounding the solution

- Getting a good guess – seems like a chicken and egg problem – but not so.
- GRAPH the functions to get an idea!! This is the best way to get an idea of the solution and get an approximate solution. In case of multiple solutions it also ensures that you have the solution that you want! *People who omit the sketch throw away the most important problem solving tool they possess – geometric intuition.*
- Incremental search – ie start with different guesses and search.
- Past experience with the problem or a similar one.

Bounding the solution

- Getting a good guess – seems like a chicken and egg problem – but not so.
- GRAPH the functions to get an idea!! This is the best way to get an idea of the solution and get an approximate solution. In case of multiple solutions it also ensures that you have the solution that you want! *People who omit the sketch throw away the most important problem solving tool they possess – geometric intuition.*
- Incremental search – ie start with different guesses and search.
- Past experience with the problem or a similar one.
- Solution of a simplified approximate model.

Bounding the solution

- Getting a good guess – seems like a chicken and egg problem – but not so.
- GRAPH the functions to get an idea!! This is the best way to get an idea of the solution and get an approximate solution. In case of multiple solutions it also ensures that you have the solution that you want! *People who omit the sketch throw away the most important problem solving tool they possess – geometric intuition.*
- Incremental search – ie start with different guesses and search.
- Past experience with the problem or a similar one.
- Solution of a simplified approximate model.
- Previous solution in a sequence of solutions.

Iterative refinement of the solution

- Iterative refining the solution involves determining the solution to a specified tolerance by a systematic procedure.

Iterative refinement of the solution

- Iterative refining the solution involves determining the solution to a specified tolerance by a systematic procedure.
- *There are numerous pitfalls in finding the roots of non linear equations*.

Iterative refinement of the solution

- Iterative refining the solution involves determining the solution to a specified tolerance by a systematic procedure.
- *There are numerous pitfalls in finding the roots of non linear equations*.
- An important question is when to stop the iteration –

$$\text{Absolute error:} |f_{i+1} - f_i|$$

$$\text{Relative error:} |\frac{f_{i+1} - f_i}{f_{i+1}}|$$

Types of Methods

There are two types of methods for finding roots of non linear equations:

- Closed domain (bracketing) methods.
- Open domain (non-bracketing) methods.

Closed domain (bracketing) method

- These are methods that start with two values of x, a and b, which bracket the root in the interval $[a, b]$.

Closed domain (bracketing) method

- These are methods that start with two values of x, a and b, which bracket the root in the interval $[a, b]$.
- If $f(a)$ and $f(b)$ have opposite signs (and if the function is continuous), then there must be atleast one root in $[a, b]$.

Closed domain (bracketing) method

- These are methods that start with two values of x, a and b, which bracket the root in the interval $[a, b]$.
- If $f(a)$ and $f(b)$ have opposite signs (and if the function is continuous), then there must be atleast one root in $[a, b]$.
- Most common closed domain methods:
  - Bisection method – interval halving.
  - False position method.

Closed domain (bracketing) method

- These are methods that start with two values of x, a and b, which bracket the root in the interval $[a, b]$.
- If $f(a)$ and $f(b)$ have opposite signs (and if the function is continuous), then there must be atleast one root in $[a, b]$.
- Most common closed domain methods:
  - Bisection method – interval halving.
  - False position method.
- In general, bracketing methods are quite robust – ie they are guaranteed to give a solution as the solution is bracketted in the interval.

Bisection Method

This is the simplest and the most robust method!

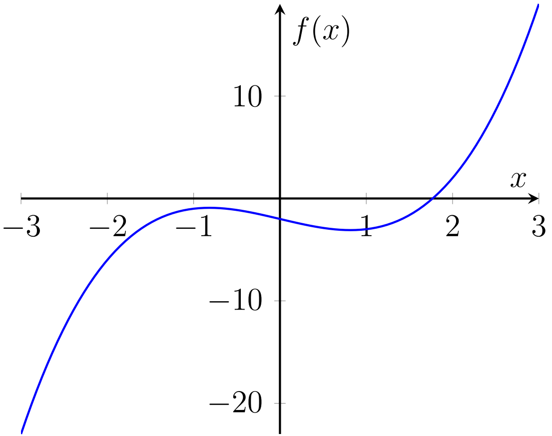
- $f(x)$ a continuous function on $[a, b]$.
- $f(x)$ changes sign between a and b ie $f(a)f(b) < 0$.

Bisection Method

This is the simplest and the most robust method!

- $f(x)$ a continuous function on $[a, b]$.
- $f(x)$ changes sign between a and b ie $f(a)f(b) < 0$.

Consider an example: $f(x) = x^3 - 2x - 2$



Bisection Method: Algorithm

- Divide $[a, b]$ into two equal parts with $c = \frac{a+b}{2}$.

Bisection Method: Algorithm

- Divide $[a, b]$ into two equal parts with $c = \frac{a+b}{2}$.
-

$$f(a)f(c) \begin{cases} < 0 \text{ there is a root in } [a, c] \implies a = a, b = c \\ > 0 \text{ there is a root in } [b, c] \implies a = c, b = b \\ = 0 \text{ root is at } c \end{cases}$$

Bisection Method: Algorithm

- Divide $[a, b]$ into two equal parts with $c = \frac{a+b}{2}$.
-

$$
f(a)f(c) \begin{cases} < 0 \text{ there is a root in } [a, c] \implies a = a, b = c \\ > 0 \text{ there is a root in } [b, c] \implies a = c, b = b \\ = 0 \text{ root is at } c \end{cases}
$$

- Interval halving is an iterative procedure.

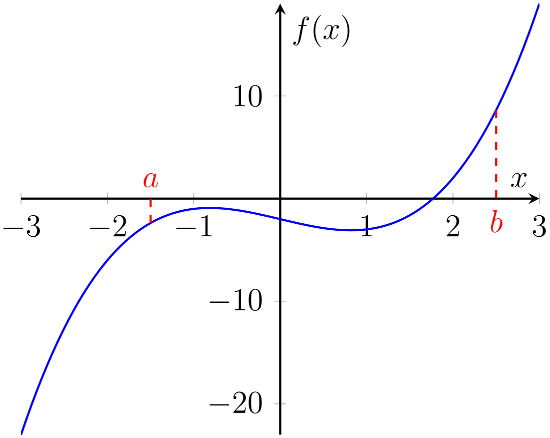
Bisection Method: Algorithm

- Divide $[a, b]$ into two equal parts with $c = \frac{a+b}{2}$.
-

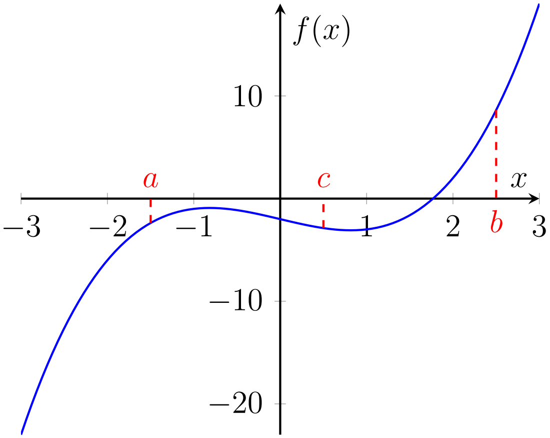
$$
f(a)f(c) \begin{cases} < 0 \text{ there is a root in } [a, c] \implies a = a, b = c \\ > 0 \text{ there is a root in } [b, c] \implies a = c, b = b \\ = 0 \text{ root is at } c \end{cases}
$$

- Interval halving is an iterative procedure.
- Continue iterations till $|b - a| < \text{tol}$ or $|f(c)| < \text{tol}$ or both

Bisection Method: Example



Bisection Method: Example



Bisection Method: analysis

- Each iteration reduces the original interval $[a, b]$ by a factor of 2. After n iterations, the size of the interval is $\frac{(b-a)}{2^n}$.

Bisection Method: analysis

- Each iteration reduces the original interval $[a, b]$ by a factor of 2. After n iterations, the size of the interval is $\frac{(b-a)}{2^n}$.
- As a result, the root remains bracketed and the method is guaranteed to converge.

Bisection Method: analysis

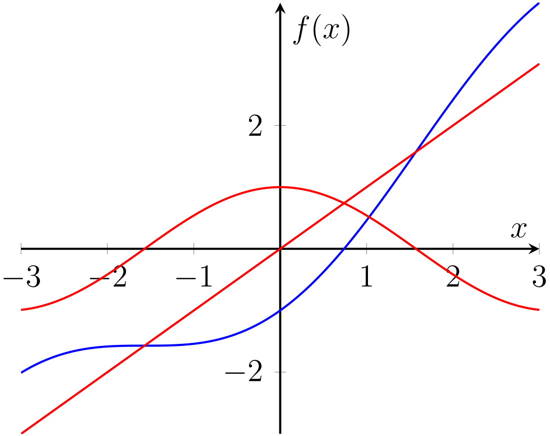
- Each iteration reduces the original interval $[a, b]$ by a factor of 2. After n iterations, the size of the interval is $\frac{(b-a)}{2^n}$.
- As a result, the root remains bracketed and the method is guaranteed to converge.
- However, the convergence of the method can be quite slow.

Bisection Method: analysis

- Each iteration reduces the original interval $[a, b]$ by a factor of 2. After n iterations, the size of the interval is $\frac{(b-a)}{2^n}$.
- As a result, the root remains bracketed and the method is guaranteed to converge.
- However, the convergence of the method can be quite slow.
- Notice that this method does not use any information about the function's behaviour!

Bisection method: Example

Consider an example: $f(x) = x - \cos(x)$



Bisection method: example

i	a	f(a)	b	f(b)	c	f(c)
1	0.000000	-1.000000	4.000000	4.653644	2.000000	2.416147
2	0.000000	-1.000000	2.000000	2.416147	1.000000	0.459698
3	0.000000	-1.000000	1.000000	0.459698	0.500000	-0.377583
4	0.500000	-0.377583	1.000000	0.459698	0.750000	0.018311
5	0.500000	-0.377583	0.750000	0.018311	0.625000	-0.185963
6	0.625000	-0.185963	0.750000	0.018311	0.687500	-0.085335
7	0.687500	-0.085335	0.750000	0.018311	0.718750	-0.033879
8	0.718750	-0.033879	0.750000	0.018311	0.734375	-0.007875
9	0.734375	-0.007875	0.750000	0.018311	0.742188	0.005196
10	0.734375	-0.007875	0.742188	0.005196	0.738281	-0.001345
11	0.738281	-0.001345	0.742188	0.005196	0.740234	0.001924
12	0.738281	-0.001345	0.740234	0.001924	0.739258	0.000289
13	0.738281	-0.001345	0.739258	0.000289	0.738770	-0.000528
14	0.738770	-0.000528	0.739258	0.000289	0.739014	-0.000120
15	0.739014	-0.000120	0.739258	0.000289	0.739136	0.000085
16	0.739014	-0.000120	0.739136	0.000085	0.739075	-0.000017
17	0.739075	-0.000017	0.739136	0.000085	0.739105	0.000034
18	0.739075	-0.000017	0.739105	0.000034	0.739090	0.000008
19	0.739075	-0.000017	0.739090	0.000008	0.739082	-0.000005
20	0.739082	-0.000005	0.739090	0.000008	0.739086	0.000002
21	0.739082	-0.000005	0.739086	0.000002	0.739084	-0.000001
22	0.739084	-0.000001	0.739086	0.000002	0.739085	0.000000

```
Root is: 0.739085197449
```

False Position Method

- In this method, the function $f(x)$ is assumed to be a linear function $g(x)$ in the interval $[a, b]$.

False Position Method

- In this method, the function $f(x)$ is assumed to be a linear function $g(x)$ in the interval $[a, b]$.
- The root of the linear function $g(x), x = c$ is taken to be the next approximation of the root of the original function $f(x)$.

False Position Method

- In this method, the function $f(x)$ is assumed to be a linear function $g(x)$ in the interval $[a, b]$.
- The root of the linear function $g(x), x = c$ is taken to be the next approximation of the root of the original function $f(x)$.
- The root of the linear function, $g(x)$, $x = c$ is not the root of the non linear function $f(x)$. It is a *false position* and hence the name.

False Position Method

- In this method, the function $f(x)$ is assumed to be a linear function $g(x)$ in the interval $[a, b]$.
- The root of the linear function $g(x), x = c$ is taken to be the next approximation of the root of the original function $f(x)$.
- The root of the linear function, $g(x)$, $x = c$ is not the root of the non linear function $f(x)$. It is a *false position* and hence the name.
- Unlike bisection, this method uses some information about the function $f(x)$.

False position method: Algorithm

The slope of linear function, $g'(x)$, is given by:

$$g'(x) = \frac{f(b) - f(a)}{b - a}$$

Assuming $f(c) = 0$, one can also write $g'(x)$ as:

$$g'(x) = \frac{f(b) - f(c)}{b - c} \implies c = b - \frac{f(b)}{g'(x)}$$

Combining with the first equation:

$$c = b - f(b) * \frac{b - a}{f(b) - f(a)}$$

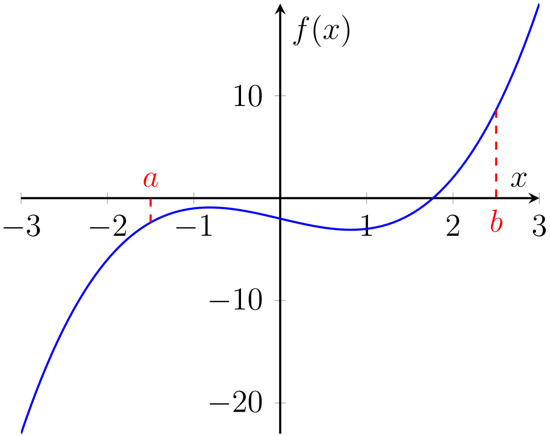
$$c = \frac{a * f(b) - b * f(a)}{f(b) - f(a)}$$

Then just like the bisection method:

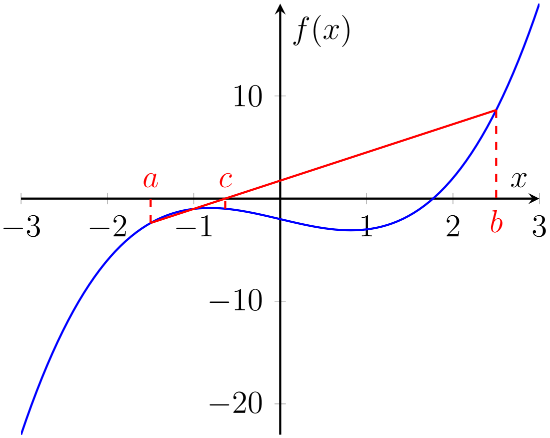
$$\text{if } f(a) * f(c) < 0 \; a = a, b = c$$

$$\text{if } f(a) * f(c) > 0 \; a = c, b = b$$

False position method: Example



False position method: Example



False position method: example

i	a	f(a)	b	f(b)	c	f(c)
1	0.000000	-1.000000	4.000000	4.653644	0.707508	-0.052475
2	0.707508	-0.052475	4.000000	4.653644	0.744221	0.008605
3	0.707508	-0.052475	0.744221	0.008605	0.739049	-0.000061
4	0.739049	-0.000061	0.744221	0.008605	0.739085	-0.000000

Root is: 0.739085092149

False position used 4 iterations compared to 22 in bisection!
Generally false position converges much faster than bisection.