Bisection method ***Lec-01***

The bisection method is simple, robust, and straight-forward: take an interval [a, b] such that f(a) and f(b) have opposite signs, find the midpoint of [a, b], and then decide whether the root lies on [a, (a + b)/2] or [(a + b)/2, b]. Repeat until the interval is sufficiently small.

Initial Requirements

We have an initial bound [a, b] on the root, that is, f(a) and (b) have opposite signs.

Iteration Process

Given the interval [a, b], define $c = (a + b)/2$. Then

- if f(c) = 0 (unlikely in practice), then halt, as we have found a root,
- if f(c) and f(a) have opposite signs, then a root must lie on [a, c], so assign $b = c$,
- else f(c) and f(b) must have opposite signs, and thus a root must lie on [c, b], so assign $a = c$.

Example 1

Consider finding the root of $f(x) = x^2 - 3$ and start with the interval [1, 2].

Table 1. Bisection method applied to $f(x) = x^2 - 3$.

a	***b***	**f(*a*)**	**f(*b*)**	***c* = (a + b)/2**	**f(*c*)**	**Update**
1.0	2.0	-2.0	1.0	1.5	-0.75	a = c
1.5	2.0	-0.75	1.0	1.75	0.062	b = c
1.5	1.75	-0.75	0.0625	1.625	-0.359	a = c
1.625	1.75	-0.3594	0.0625	1.6875	-0.1523	a = c
1.6875	1.75	-0.1523	0.0625	1.7188	-0.0457	a = c
1.7188	1.75	-0.0457	0.0625	1.7344	0.0081	b = c

1.71988/td>	1.7344	-0.0457	0.0081	1.7266	-0.0189	a = c

Thus, with the seventh iteration, we note that the final interval, [1.7266, 1.7344], has a width less than 0.01 and $|f(1.7344)| < 0.01$, and therefore we chose b = 1.7344 to be our approximation of the root

Example 3

Approximate the root of $f(x) = x^3 - 3$ with the bisection method starting with the interval [1, 2]

Answer: 1.4375

Example 4

Approximate the root of $f(x) = x^2 - 10$ with the bisection method starting with the interval [3, 4]

Answer: 3.15625

Lec-02

Example 5

Consider finding the root of $f(x) = e^{-x}(3.2 \sin(x) - 0.5 \cos(x))$ on the interval [3, 4]

Table 1. Bisection method applied to $f(x) = e^{-x}(3.2 \sin(x) - 0.5 \cos(x))$.

a	***b***	**f(*a*)**	**f(*b*)**	***c* = (a + b)/2**	**f(*c*)**	**Update**	**new b − a**
3.0	4.0	0.047127	-0.038372	3.5	-0.019757	b = c	0.5
3.0	3.5	0.047127	-0.019757	3.25	0.0058479	a = c	0.25
3.25	3.5	0.0058479	-0.019757	3.375	-0.0086808	b = c	0.125
3.25	3.375	0.0058479	-0.0086808	3.3125	-0.0018773	b = c	0.0625
3.25	3.3125	0.0058479	-0.0018773	3.2812	0.0018739	a = c	0.0313
3.2812	3.3125	0.0018739	-0.0018773	3.2968	-0.000024791	b = c	0.0156
3.2812	3.2968	0.0018739	-0.000024791	3.289	0.00091736	a = c	0.0078
3.289	3.2968	0.00091736	-0.000024791	3.2929	0.00044352	a = c	0.0039
3.2929	3.2968	0.00044352	-0.000024791	3.2948	0.00021466	a = c	0.002

3.2948	3.2968	0.00021466	-0.000024791	3.2958	0.000094077	a = c	0.001
3.2958	3.2968	0.000094077	-0.000024791	3.2963	0.000034799	a = c	0.0005

Thus, after the 11th iteration, we note that the final interval, [3.2958, 3.2968] has a width less than 0.001 and $|f(3.2968)| < 0.001$ and therefore we chose b = 3.2968 to be our approximation of the root.

Lec-03

The False-Position Method

Introduction

The false-position method is a modification on the bisection method: if it is known that the root lies on [*a*, *b*], then it is reasonable that we can approximate the function on the interval by interpolating the points (*a*, f(*a*)) and (*b*, f(*b*)). In that case, why not use the root of this linear interpolation as our next approximation to the root?

Initial Requirements

We have an initial bound [*a*, *b*] on the root, that is, f(*a*) and f(*b*) have opposite signs.

Iteration Process

Given the interval [*a*, *b*], define $c = (a\,f(b) - b\,f(a))/(f(b) - f(a))$. Then

- if f(*c*) = 0 (unlikely in practice), then halt, as we have found a root,
- if f(*c*) and f(*a*) have opposite signs, then a root must lie on [*a*, *c*], so assign *step* = *b* - *c* and assign *b* = *c*,
- else f(*c*) and f(*b*) must have opposite signs, and thus a root must lie on [*c*, *b*], so assign *step* = *c* - *a* and assign *a* = *c*.

Halting Conditions

There are three conditions which may cause the iteration process to halt:

1. As indicated, if f(*c*) = 0.
2. We halt if both of the following conditions are met:
   - The step size is sufficiently small, that is $step < \varepsilon_{step}$, and
   - The function evaluated at one of the end point $|f(a)|$ or $|f(b)| < \varepsilon_{abs}$.
3. If we have iterated some maximum number of times, say *N*, and have not met Condition 1, we halt and indicate that a solution was not found.

If we halt due to Condition 1, we state that *c* is our approximation to the root. If we halt according to Condition 2, we choose either *a* or *b*, depending on whether $|f(a)| < |f(b)|$ or $|f(a)| > |f(b)|$, respectively.

If we halt due to Condition 3, then we indicate that a solution may not exist (the function may be discontinuous).

Example 1

Consider finding the root of $f(x) = x^2 - 3$. start with the interval [1, 2].

Table 1. False-position method applied to $f(x) = x^2 - 3$.

a	***b***	**f(*a*)**	**f(*b*)**	***C***=(*a* f(*b*) − *b* f(*a*))/(f(*b*) − f(*a*)).	**f(*c*)**	**Update**
1.0	2.0	-2.00	1.00	1.6667	-0.2221	a = c
1.6667	2.0	-0.2221	1.0	1.7273	-0.0164	a = c
1.7273	2.0	-0.0164	1.0	1.7317	0.0012	a = c

Thus, with the third iteration, we note that the last step 1.7273 → 1.7317 is less than 0.01 and $|f(1.7317)| < 0.01$, and therefore we chose b = 1.7317 to be our approximation of the root.

Example 2

Consider finding the root of $f(x) = e^{-x}(3.2 \sin(x) - 0.5 \cos(x))$ on the interval [3, 4],

-

Table 2. False-position method applied to $f(x) = e^{-x}(3.2 \sin(x) - 0.5 \cos(x))$.

a	***b***	**f(*a*)**	**f(*b*)**	***C***=(*a* f(*b*) − *b* f(*a*))/(f(*b*) − f(*a*)).	**f(*c*)**	**Update**
3.0	4.0	0.047127	-0.038372	3.5513	-0.023411	b = c
3.0	3.5513	0.047127	-0.023411	3.3683	-0.0079940	b = c
3.0	3.3683	0.047127	-0.0079940	3.3149	-0.0021548	b = c
3.0	3.3149	0.047127	-0.0021548	3.3010	-0.00052616	b = c
3.0	3.3010	0.047127	-0.00052616	3.2978	-0.00014453	b = c
3.0	3.2978	0.047127	-0.00014453	3.2969	-0.000036998	b = c

Thus, after the sixth iteration, we note that the final step, 3.2978 → 3.2969 has a size less than 0.001 and $|f(3.2969)| < 0.001$ and therefore we chose b = 3.2969 to be our approximation of the root.

Newton-Raphson Technique

The Newton-Raphson method is one of the most widely used methods for root finding. It can be easily generalized to the problem of finding solutions of a system of non-linear equations, which is referred to as Newton's technique. Moreover, it can be shown that the technique is quadratically convergent as we approach the root.

Unlike the bisection and false position methods, the Newton-Raphson (N-R) technique requires only one inital value x_0, which we will refer to as the *initial guess* for the root. To see how the N-R method works, we can rewrite the function $f(x)$ using a Taylor series expansion in $(x\text{-}x_0)$:

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + 1/2\, f''(x_0)(x-x_0)^2 + \ldots = 0 \qquad (5)$$

where $f'(x)$ denotes the first derivative of $f(x)$ with respect to x, $f''(x)$ is the second derivative, and so forth. Now, suppose the initial guess is pretty close to the real root. Then $(x\text{-}x_0)$ is small, and only the first few terms in the series are important to get an accurate estimate of the true root, given x_0. By truncating the series at the second term (linear in x), we obtain the N-R iteration formula for getting a better estimate of the true root:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \qquad (6)$$

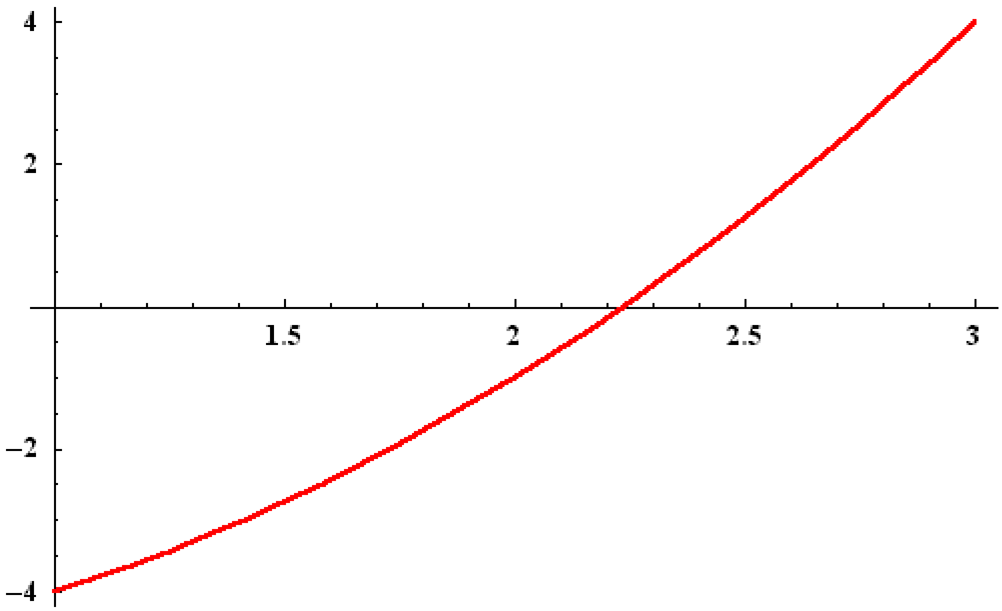
Thus the N-R method finds the tangent to the function $f(x)$ at $x=x_0$ and extrapolates it to intersect the x axis to get x_1. This point of intersection is taken as the new approximation to the root and the procedure is repeated until convergence is obtained whenever possible. Mathematically, given the value of $x = x_i$ at the end of the ith iteration, we obtain x_{i+1} as

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad (7)$$

We assume that the derivative does not vanish for any of the $x_{k,\ k=0,1,\ldots,\ i+1}$. The result obtained from this method with $x_0 = 0.1$

Example. Let us find an approximation to $\sqrt{5}$ to ten decimal places on the interval [1, 3].

Note that $\sqrt{5}$ is an irrational number. Therefore the sequence of decimals which defines $\sqrt{5}$ will not stop. Clearly $r = \sqrt{5}$ is the only zero of $f(x) = x^2 - 5$ on the interval [1, 3]. See the Picture.



Let $\{x_n\}$ be the successive approximations obtained through Newton's method. We have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - 5}{2x_n}\,.$$

Let us start this process by taking $x_1 = 2$.

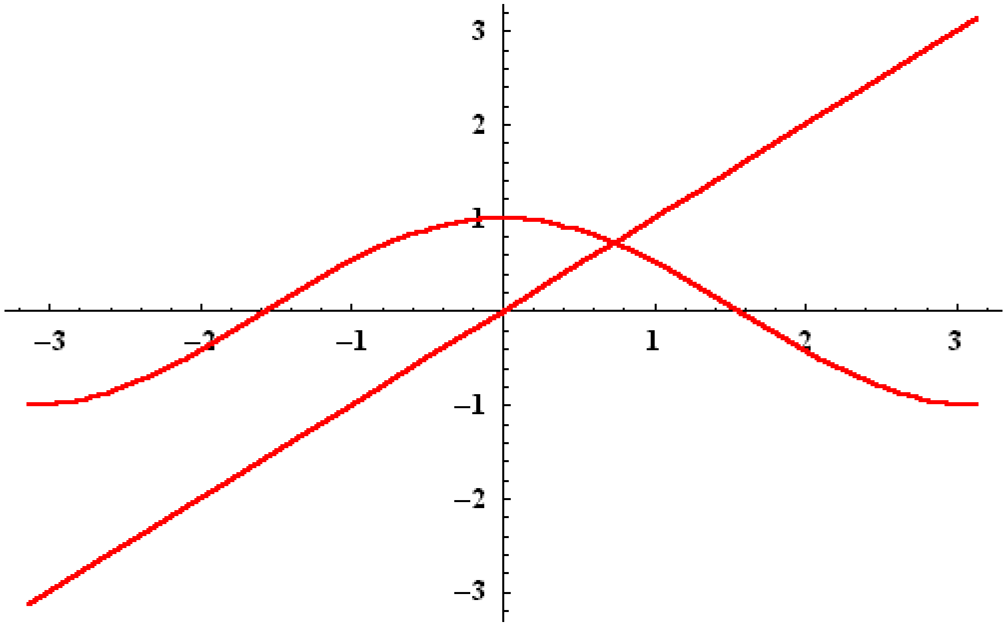
$$\begin{aligned}
x_1 &= 2\\
x_2 &= 2.25\\
x_3 &= 2.236111111111111111111111111111111\\
x_4 &= 2.236067977915804002760524499654934\\
x_5 &= 2.236067977499789696447872828327110\\
x_6 &= 2.236067977499789696409173668731276
\end{aligned}$$

It is quite remarkable that the results stabilize for more than ten decimal places after only 5 iterations!

Example. Let us approximate the only solution to the equation on the interval [0, π/2]

$$x = \cos(x)\,.$$

In fact, looking at the graphs we can see that this equation has one solution.



This solution is also the only zero of the function $f(x) = x - \cos(x)$. So now we see how Newton's method may be used to approximate *r*. Since *r* is between 0 and $\frac{\pi}{2}$, we set $x_1 = 1$. The rest of the sequence is generated through the formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n - \cos(x_n)}{1 + \sin(x_n)}\ .$$

We have

$$\begin{aligned}
x_1 &= 1. \\
x_2 &= 0.750363867840243893034942306682177 \\
x_3 &= 0.739112890911361670360585290904890 \\
x_4 &= 0.739085133385283969760125120856804 \\
x_5 &= 0.739085133215160641661702625685026 \\
x_6 &= 0.739085133215160641655312087673873 \\
x_7 &= 0.739085133215160641655312087673873 \\
x_8 &= 0.739085133215160641655312087673873
\end{aligned}$$

Exercise 1. Approximate the real root in the interval to four decimal places of

$$x^3 + 5x - 3 = 0\ .$$

Exercise 2. Approximate to four decimal places $\sqrt[3]{3}$. in the interval [1, 3]

Example:

Solve $5\sin^2 x - 8\cos^5 x = 0$ in [0.5,1.5] for the root by Newton-Raphson method.

Solution: Given

$$f(x) = 5\sin^2 x - 8cos^5 x$$

$$f'(x) = 10\sin x\cos x + 40cos^4 x\sin x$$

Say, $x_0 = 0.5, \quad \epsilon = 10^{-6}$

The results are tabulated below:

Newton Raphson Method

Iteration no.	x_n	x_{n+1}	$\lvert f(x_{n+1})\rvert$
0	0.5000000000	0.6934901476	0.1086351126
1	0.6934901476	0.7013291121	0.0005313741
2	0.7013291121	0.7013678551	0.0000003363

FIXED POINT ITERATION METHOD

Fixed point **:** A point, say, **s** is called a fixed point if it satisfies the equation **x = g(x)**.

Fixed point Iteration **:** The transcendental equation **f(x) = 0** can be converted algebraically into the form **x = g(x)** and then using the iterative scheme with the recursive relation

$$x_{i+1} = g(x_i), \qquad i = 0, 1, 2, \ldots,$$

with some initial guess x_0 is called the fixed point iterative scheme.

Algorithm - Fixed Point Iteration Scheme

Given an equation f(x) = 0
Convert f(x) = 0 into the form x = g(x)
Let the initial guess be x_0

Do
    $x_{i+1}= g(x_i)$
while (none of the convergence criterion C1 or C2 is met)

- C1. Fixing apriori the total number of iterations **N** .
- C2. By testing the condition **$| x_{i+1} - g(x_i) |$** (where **i** is the iteration number) less than some tolerance limit, say epsilon, fixed apriori.

Numerical Example :

Find a root of **$x^4-x-10 = 0$**

Consider another function **$g2(x) = (x + 10)^{1/4}$** and the fixed point iterative scheme
$x_{i+1}= (x_i + 10)^{1/4}$, $i = 0, 1, 2, . . .$

let the initial guess **x_0** be **1.0, 2.0 and 4.0**

i	0	1	2	3	4	5	6
x_i	1.0	1.82116	1.85424	1.85553	1.85558	**1.85558**	
x_i	2.0	1.861	1.8558	1.85559	1.85558	**1.85558**	
x_i	4.0	1.93434	1.85866	1.8557	1.85559	1.85558	**1.85558**

That is for **g2** the iterative process is converging to **1.85558 .**

Consider **$g3(x) =(x+10)^{1/2}/x$** and the fixed point iterative scheme

$x_{i+1}=(x_i+10)^{1/2} /x_i$, $i = 0, 1, 2, . . .$
let the initial guess **x_0** be **1.8,**

i	0	1	2	3	4	5	6	. . .	98
x_i	1.8	1.9084	1.80825	1.90035	1.81529	1,89355	1.82129	. . .	**1.8555**

That is for g3 with any initial guess the iterative process is converging but very slowly to

Geometric interpretation of convergence with g2 and g3

******** Consider **$g1(x) = 10 / (x^3-1)$** and the fixed point iterative scheme **$x_{i+1}=10 / (x_i^3 -1)$, $i = 0, 1, 2, . .$** let the initial guess **x_0** be **2.0**

i	0	1	2	3	4	5	6	7	8
x_i	2	1.429	5.214	0.071	-10.004	-9.978E-3	-10	-9.99E-3	-10

So the iterative process with **g1** gone into an infinite loop without converging.

1. Find the root of **(cos[x])-(x * exp[x]) = 0**

Consider **g(x) = cos[x]/exp[x]**

The graph of **g(x)** and **x** are given in the figure.

let the initial guess $\mathbf{x_0}$ be **2.0**

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...	31	32
x_i	1	0.199	0.803	0.311	0.698	0.381	0.634	0.427	0.594	0.458	0.567	0.478	0.551	0.491	...	0.518	**0.518**

That is for **g(x) = cos[x]/exp[x]** the itirative process is converged to **0.518.**

2. Find the root of $\mathbf{x^4-x-10 = 0}$

Consider **g(x) =** $\mathbf{(x + 10)^{1/4}}$

The graph of **g(x)** and **x** are given in the figure.

let the initial guess $\mathbf{x_0}$ be **4.0**

i	0	1	2	3	4	5	6
x_i	4.0	1.93434	1.85866	1.8557	1.85559	1.85558	**1.85558**

That is for **g(x) =** $\mathbf{(x + 10)^{1/4}}$ the itirative process is converged to **1.85558**.

3. Find the root of **x-sin[x]-(1/2)= 0**

Consider **g(x) = sin[x]+(1/2)**

The graph of **g(x)** and **x** are given in the figure.

let the initial guess $\mathbf{x_0}$ be **2**

i	0	1	2	3	4	5
x_i	2	1.409	1.487	1.496	1.497	**1.497**

That is for **g(x) = sin[x]+(1/2)** the itirative process is converged to **1.497**.

4. Find the root of **exp[-x]=3log[x]**

Consider **g(x) = exp[(exp[-x]/3)]**

The graph of **g(x)** and **x** are given in the figure.

let the initial guess $\mathbf{x_0}$ be **2**

i	0	1	2	3	4	5
x_i	2	1.046	1.124	1.114	1.116	**1.115**

That is for **g(x) = exp[(exp[-x]/3)]** the itirative process is converged to **1.115**.