Numerical Analysis MTH614

Spring 2012, Korea University

Finding roots

Bisection method

We begin to study a set of root-finding methods starting form the simplest one, the Bisection Method.

The Bisection method approximates the root of an equation on an interval by iteratively halving the interval until the desired solution is reached.

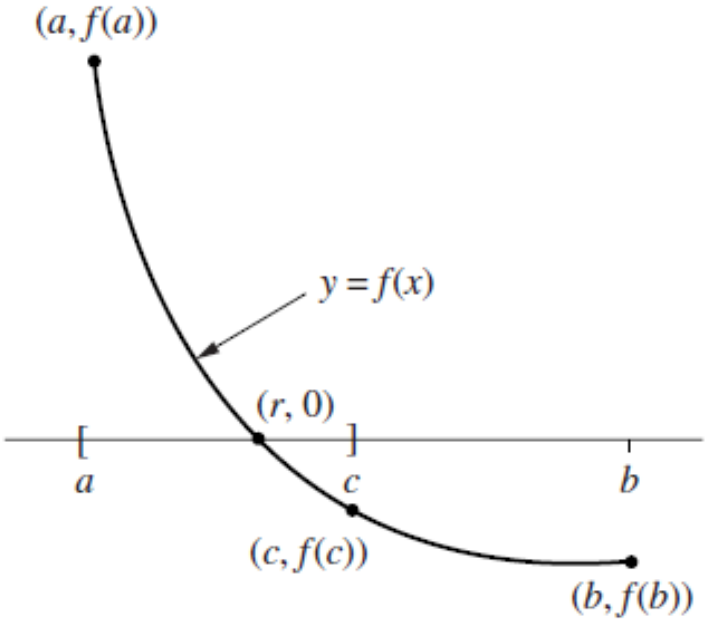
In other word, when a function sign is changed, the location of the root is then determined as lying within the subinterval where the sign change occurs. The subinterval then becomes the interval for the next iteration. The process is repeated until the required precision solution is found.

Note that the Bisection method operates under the conditions necessary for the intermediate value theorem.

Intermediate value theorem

Suppose $y = f(x) \in C,\ x \in [a, b] \subset \mathbb{R}$ and $f(a)f(b) < 0$, then there is $r \in (a, b)$ such that $r \in [a, b]$.

Note that the root $f(r) = 0$ found is not necessarily unique.



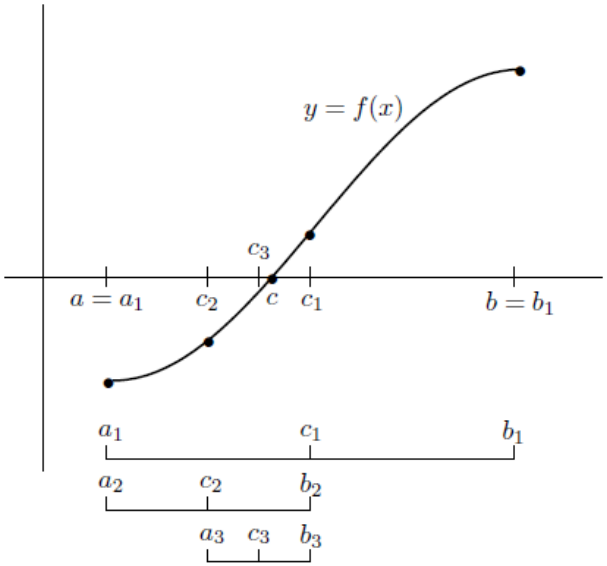
1. Let $\epsilon > 0$ be the upper bound for the error required of the numerical solution.

2. Compute $c = \dfrac{a+b}{2}$ and $d = f(c) \cdot f(a)$.

3. If $d < 0$, then let $b = c$ and $a = a$. If $d > 0$ then $a = c$ and $b = b$. If $|d| < \epsilon$ then c is a solution of $f(x) = 0$.

4. The new interval $[a, b]$ will be then be half the length of the original interval.

Repeat 2, 3, and 4 until the numerical solution is found.



For example, we consider a continuous function such as

$$f(x) = x^5 + 3x - 1 = 0$$

then, the root of the function can be found by the following MATLAB code

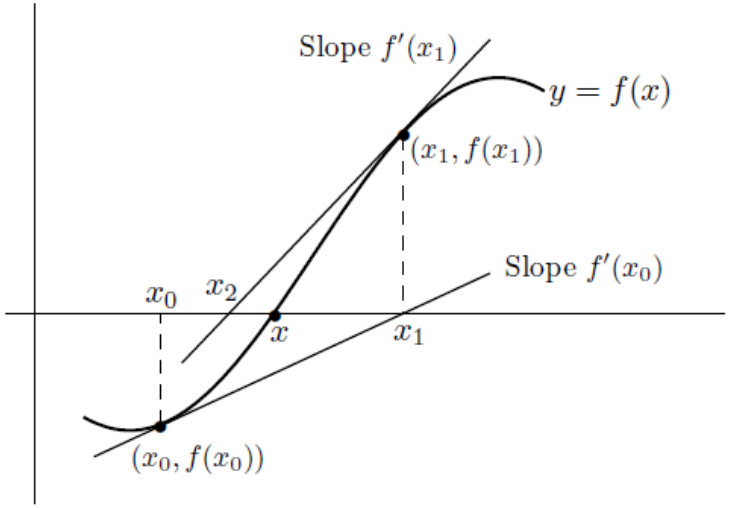
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% bisection.m %%%%%%%%%%%%%%%%%%%%%%
clear; clc; clf;
f='x^5+3*x-1';
n=1; a=0; b=1; c=(a+b)/2; tol=0.001;
fprintf(' n       a             b           c          b-c \n')
while b-c>=tol
 n=n+1;
 x=b; fb=eval(f);
 x=c; fc=eval(f);
 if fb*fc<=0
     a=c; c=(a+b)/2;
 else
     b=c; c=(a+b)/2;
 end
 fprintf('%2.0f %2.10f %2.10f %2.10f %2.4e \n',n,a,b,c,b-c)
end
```

n	a	b	c	b-c
2	0.0000000000	0.5000000000	0.2500000000	2.5000e-001
3	0.2500000000	0.5000000000	0.3750000000	1.2500e-001
4	0.2500000000	0.3750000000	0.3125000000	6.2500e-002
5	0.3125000000	0.3750000000	0.3437500000	3.1250e-002
6	0.3125000000	0.3437500000	0.3281250000	1.5625e-002
7	0.3281250000	0.3437500000	0.3359375000	7.8125e-003
8	0.3281250000	0.3359375000	0.3320312500	3.9063e-003
9	0.3281250000	0.3320312500	0.3300781250	1.9531e-003
10	0.3300781250	0.3320312500	0.3310546875	9.7656e-004

Newton method

The most widely used of all root finding methods is the Newton method.

We consider the technique briefly.
The Newton method can be derived on the basis of this geometrical interpretation.
As in the below figure, the first derivative at x is equivalent to the

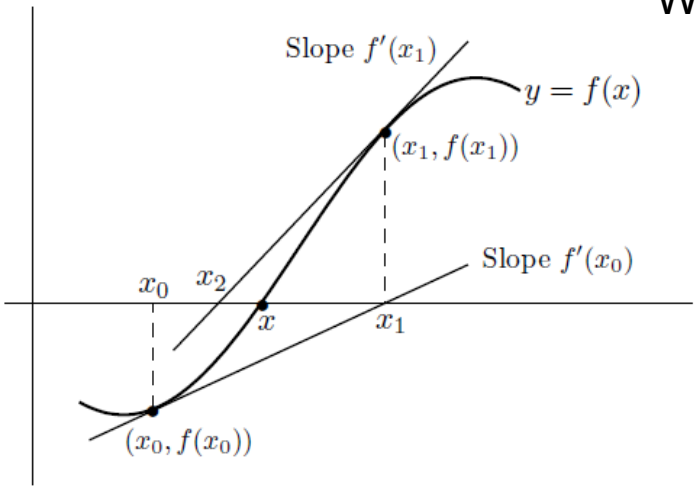


tangent line.

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \qquad (1)$$

which can be rearranged to yield

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \, . \qquad (2)$$



With this equation, we generalize that

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \quad (1)$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (2)$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \quad (3)$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \ldots \quad (4)$$

This equation is called the Newton formula.

Use the Newton method to estimate the root of

$$f(x) = x^5 + 3x - 1 = 0$$

employing an initial guess of $x_0 = 0.5$.

```
%%%%%%%%%%%%%%%%%%%%%%%%%% newton.m %%%%%%%%%%%%%%%%%%%%%%%%%%
clear; clc; clf;
f='x^5+3*x-1'; df='5*x^4+3';
n=0; x0=0.5; tol=10^(-10); m=1;
fprintf(' n        x_n         f(x_n)       x_n-x_(n-1) \n')
while m>=tol
 n=n+1; x=x0; y=eval(f); dy=eval(df);
 x1=x0-y/dy; m=abs(x1-x0);
 x=x1; y=eval(f);
 fprintf('%2.0f   %2.4e   %2.4e   %2.4e \n',n,x1,y,x1-x0)
 x0=x1;
end
```

n	x_n	f(x_n)	x_n-x_(n-1)
1	3.3962e-001	2.3386e-002	-1.6038e-001
2	3.3200e-001	2.2278e-005	-7.6263e-003
3	3.3199e-001	1.9385e-011	-7.2785e-006
4	3.3199e-001	2.2204e-016	-6.3335e-012

Secant method

In implementing the Newton method, there are some issues. One of them is the evaluation of the derivative. This occurs when derivatives may be difficult or inconvenient to evaluate.

For these cases, the derivative can be approximated by a back-ward finite divided difference:

$$\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \tag{1}$$

This approximation can be substituted to yield the following iterative equation

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}, \quad n = 1, 2, 3, \ldots \tag{2}$$

This is the formula for the secant method.

$$f(x) = x^5 + 3x - 1 = 0$$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% secant.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear; clc;
f='x^5+3*x-1';
n=0; x0=0; x1=1; tol=10^(-6);
fprintf(' n     x_(n+1)       f(x_(n+1))     x_(n+1)-x_n \n')
while abs(x1-x0)>=tol
 n=n+1; x=x0; y0=eval(f); x=x1; y1=eval(f);
 x2=x1-(y1*(x1-x0)/(y1-y0));
 x=x2; y=eval(f);
 fprintf('%2.0f   %5.4e   %5.4e   %5.4e \n',n,x2,y,x2-x1)
 x0=x1; x1=x2;
end
```

n	x_(n+1)	f(x_(n+1))	x_(n+1)-x_n
1	2.5000e-001	-2.4902e-001	-7.5000e-001
2	3.0748e-001	-7.4799e-002	5.7484e-002
3	3.3216e-001	5.3412e-004	2.4679e-002
4	3.3199e-001	-1.4581e-006	-1.7498e-004
5	3.3199e-001	-3.0434e-011	4.7639e-007