Information Technology

Numerical Methods

Module 4: Iterative Methods to solve equation *f* (x) = 0: Secant Method

4.1 Introduction

If we revisit our previous module, which was on method of false position, we found that though False Position Method is generally faster (there are exceptions!) than Bisection Method, it still has a linear rate of convergence. Bisection method has rate of converge $O\left(\frac{1}{2^n}\right)$, which is also linear. And so again the same question; is there a feasibility of method faster than Secant Method? So today, we would study a modification of false Position Method, called Secant Method.

4.2 Iterative Process

Secant method is an open method. That is, no more interval under consideration needs to bracket the root, though it still requires two initial guesses. Let us denote these initial guesses by x_{-1} and x_0. The formula for generating the sequence of approximations is

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

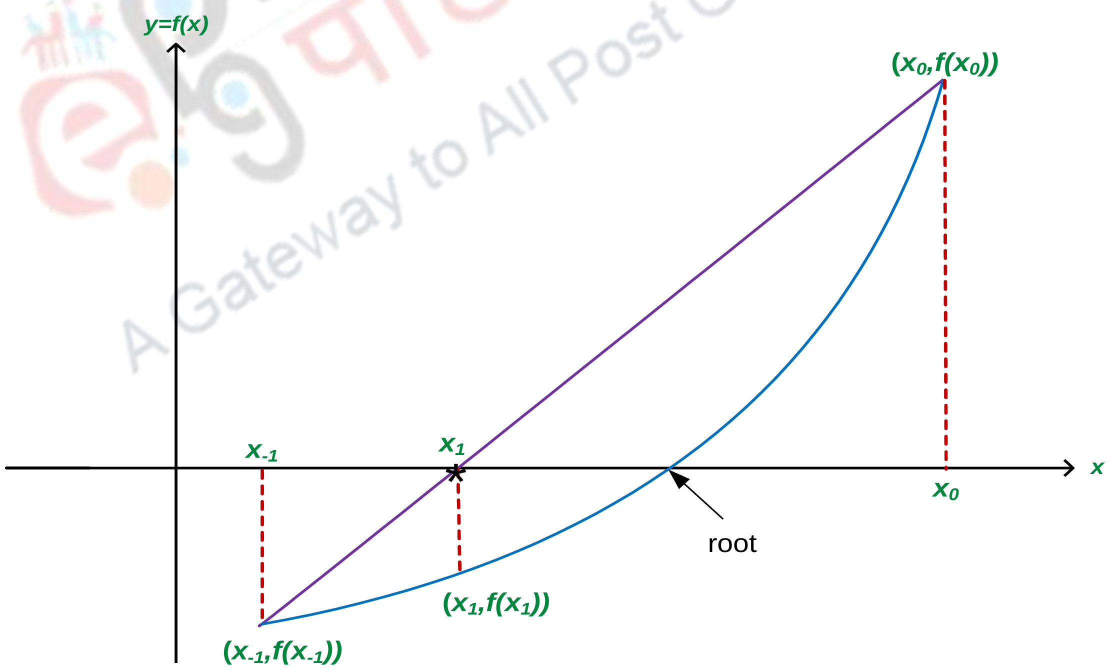
Doesn't it look same as that in method of false position? Then, what is the difference?

Well, let us examine one by one in detail:

Firstly, initial choices of x_{-1} and x_0 need not be bracketing the root. So, that is, root need not lie within the end points $[x_{-1}, x_0]$ or $[x_0, x_{-1}]$. That is $f(x_{-1})$ and $f(x_0)$ can be of same sign. No more, one needs to ensure that $f(x_{-1}).f(x_0) < 0$.

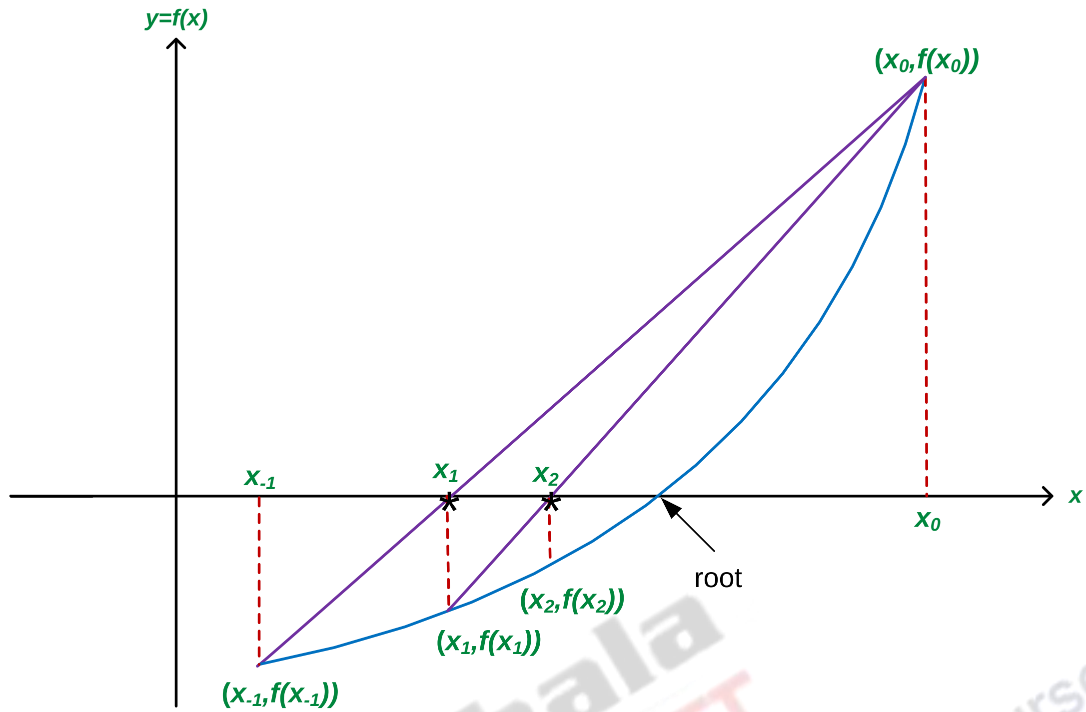
Though, in practice one usually chooses x_{-1}, x_0 as ones which bracket the root, there is no such compulsion.

Secondly, in computation of next approximation, older approximation is discarded. Recent most two approximations are used for iteration, irrespective of function values at these end points. So graphically, straight line joining $(x_{n-1}, f(x_{n-1}))$ and $(x_n, f(x_n))$ is drawn to generate x_{n+1}, next approximation. No more, function values sign are checked to ensure that root lies between $(x_{n-1}, f(x_{n-1}))$ and $(x_n, f(x_n))$. Because secant is drawn joining $(x_{n-1}, f(x_{n-1}))$ and $(x_n, f(x_n))$, it is called Secant Method. So, computationally it is even easier then method of False Position. Let us understand this method graphically.

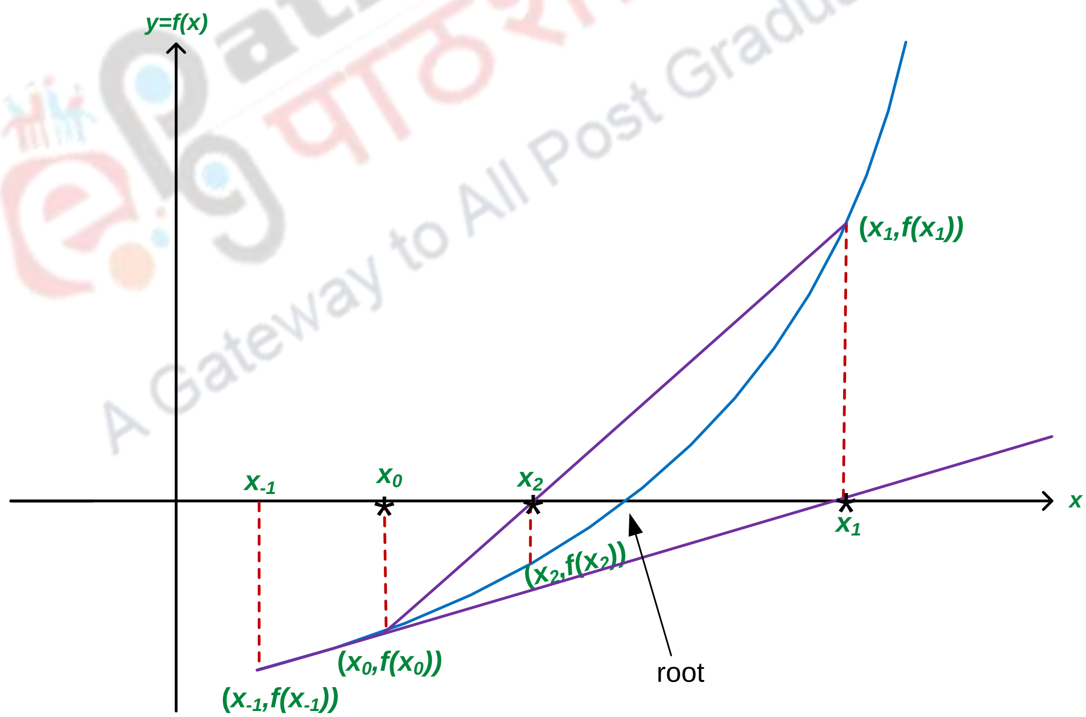


[Figure 1]

e-Pathshala पाठशाला A Gateway to All Post Graduate Courses



[Figure 2]



[Figure 3]

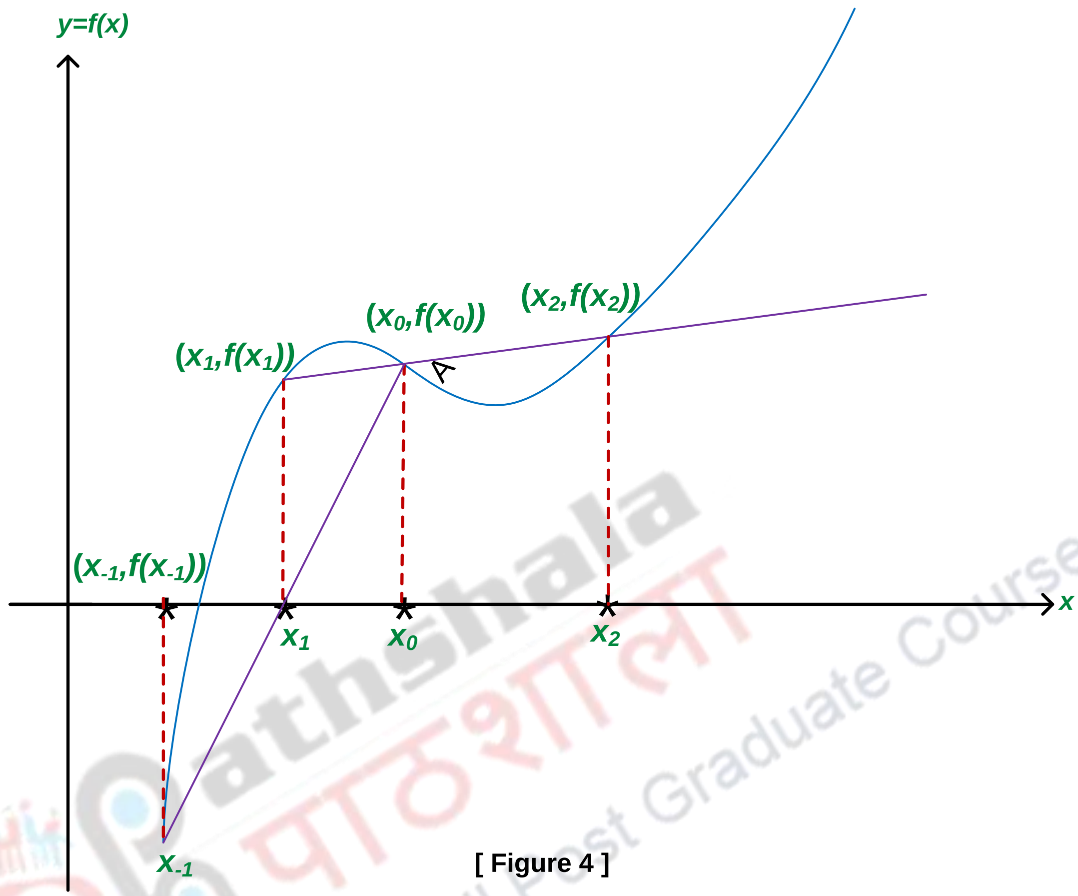
So in nutshell, process is :

1. Begin with any two guesses, x_{-1}, x_0.
2. Draw secant joining $(x_{-1}, f(x_{-1}))$ and $(x_0, f(x_0))$.
3. The point of intersection of secant with X axis is root approximation point x_1.
4. Discard x_{-1} and now draw secant joining $(x_0, f(x_0))$ and $(x_1, f(x_1))$.
5. Continue, this way, till we obtain the root correct to desired accuracy

4.3 Stopping Criterion

Any of the three stopping criterions already discussed (X-TOL, F-TOL or relative error) in the previous module on method of false position can be employed. They can be applied in combination also. Secant Method being open method does not guarantee convergence. In moving from method of false position to Secant method, we have lost guaranteed convergence. If it converges; it is quite fast as is evident from graphs and examples to follow. But, we have lost guaranteed convergence. As a result, it is always desirable and advisable in this method to keep an upper limit on number of iterations say N, be performed to avoid infinite loop or considerable delay in reaching the root .

A Gateway to All Post Graduate Courses



[Figure 4]

Moment, number of iteration exceeds value N, we should exit the loop. May be, again employ the method with better initial guesses. We shall discuss over it, today itself in this module but little later how to go about it, if the method does not converge. Thus, in conjunction with any of the stopping criterion, maximum number of iterations permitted to be performed, also would be used.

4.4 Algorithm for Secant Method

Step 1	Input $f(x)$: The given function, x_{-1}, x_0, the two initial approximations to the root $\in$ The error tolerance (X – TOL) δ The error tolerance (F – TOL) N the maximum umber of iterations
Step 2	Let n = 0
Step 3	Compute $f(x_{n-1})$ and $f(x_n)$
Step 4	Compute next approximation $x_{n+1} = \dfrac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$
Step 5	Evaluate $f(x_{n+1})$, If $f(x_{n+1})$=0; x_{n+1} is the desired root, Exit
Step 6	Test for convergence If $\lvert x_{n+1} - x_n \rvert < \in$ or/and $\lvert f(x_n) \rvert < \delta$ or $\left\lvert \dfrac{x_{n+1} - x_n}{x_n} \right\rvert < \in$
Step 7	Output: Estimate of the root is x_{n+1}, Exit
Step 8	Else, n = n +1
Step 9	If $n \leq N$, go to Step 4
Step 10	Else, output: Does not converge in N iterations

Pathshala पाठशाला A Gateway to All Post Graduate Courses

4.5 Illustrations

Let us solve the same two examples which we solved by Bisection method and method of false position, now, by Secant Method and compare the results:

Example 1:

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

$x_{-1} = 0$
$f(x_{-1}) = -1.000000$

$x_0 = 1$
$f(x_0) = 1.000000$

No.	x_{n-1}	x_n	x_{n+1}	$f(x_{n+1})$	Error $\lvert x_{n+1} - x_n \rvert$
1	0.000000	1.000000	0.500000	-0.375000	0.500000
2	1.000000	0.500000	0.636364	-0.105935	0.136364
3	0.500000	0.636364	0.690052	0.018636	0.053689
4	0.636364	0.690052	0.682020	-0.000737	0.008032
5	0.690052	0.682020	0.682326	-0.000005	0.000305
6	0.682020	0.682326	0.682328	0.000000	0.000002
7	0.682326	0.682328	0.682328	0.000000	0.000000

Desired root is 0.682328

Example 2:

$f(x_{n-1})=(3 * x) - \cos(x) - 1$

$x_{n-1} = 0$
$f(x_{n-1}) = -2.000000$

$x_n = 1$
$f(x_n) = 1.459698$

No.	x_{n-1}	x_n	x_{n+1}	$f(x_{n+1})$	Error $\lvert x_{n+1} - x_n \rvert$
1	0.000000	1.000000	0.578085	-0.103255	0.421915
2	1.000000	0.578085	0.605959	-0.004081	0.027873
3	0.578085	0.605959	0.607105	0.000014	0.001147
4	0.605959	0.607105	0.607102	0.000000	0.000004
5	0.607105	0.607102	0.607102	-0.000000	0.000000

Desired root is 0.607102

Programming Code in C:

```
//Secant Method
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include <cstdlib>
using namespace std;
#define fnx(x)(3 * x) - cos (x) -1
void main()
{
	float a,b,c,fa,fb,fc,error;
	int n=0;
	//N is Maximum Number of Iteration to take
	int N=40;
	float err=pow(10.0,-6.0)/2;

		printf("\nEnter value of a = ");
		scanf("%f", &a);
		fa = fnx(a);
		printf("\n value of f(a) = %f", fa);

		printf("\nEnter initial value of b = ");
		scanf("%f", &b);
		fb = fnx(b);
		printf("\n value of f(b) = %f", fb);
		printf("\nNo.  \t    a\t\t   b\t\t   c\t\t   f(c)\t\t   Error");

		do{
			n=n+1;
			c=(a*fb-b*fa)/(fb-fa);
			fc=fnx(c);
			error=fabs(c-b);
				printf("\n\n%d\t %f\t %f\t  %f\t %f\t %f",n,a,b,c,fc,error);
			if(error<err){
				printf("\ndesired root is %f",c);
				getch();
				exit(0);
			}
			a=b;
			fa=fb;
			b=c;
			fb=fc;
		}while(n<N);
			if(n>=N)
				printf("\n Does not converge in N iterations");
		getch();
}
```

4.5 Observations

1) Secant method is quite fast. Much faster than method of False Position.
2) In entire process, one function evaluation per iteration after starting in computation of next x_{n+1}.
3) Computationally, simple and fast, labeled as having **super linear convergence,** order of convergence is 1.618
4) Since it is open method, method may diverge in certain cases.

Example:

$f(x) = \ln x$

False Position Method
$a = 0.5$
$b = 5.0$

Secant Method
$x_{-1} = 0.5$
$x_0 = 5.0$

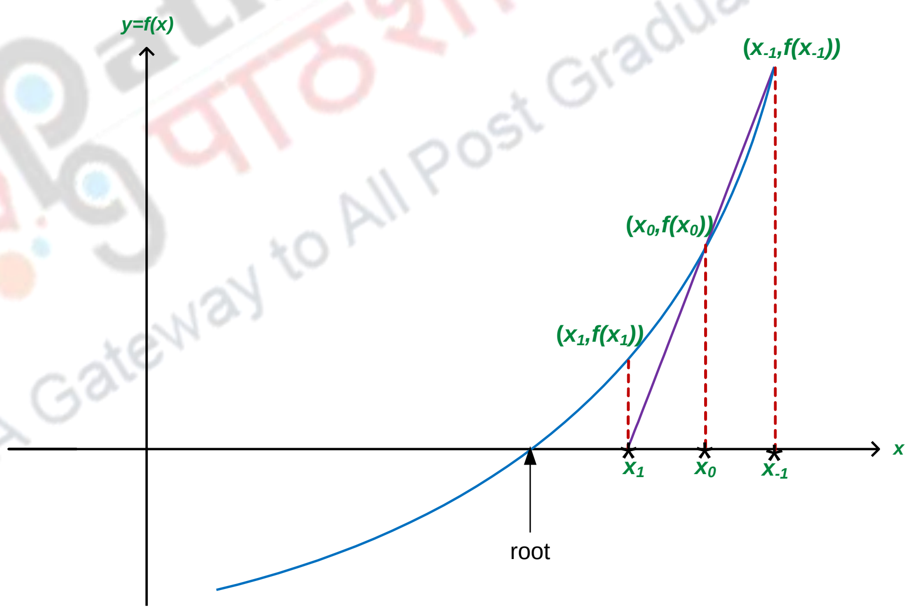
k	**False Position Method**		**Secant Method**	
	c_k	***$f(c_k)$***	***c_k***	***$f(c_k)$***
1	1.854635	0.617688	1.854635	0.617688
2	1.216308	0.195820	-0.104381	C out of domain of ln x
3	1.058521	0.056873		
4	1.016169	0.016040		
5	1.004495	0.004485		
6	1.001252	0.001251		
7	1.000349	0.000349		
8	1.000097	0.000097		
9	1.000027	0.000027		
10	1.000008	0.000008		
11	1.000002	0.000002	–	–
12	1.000001	0.000001	–	–
13	1.000000	0.000000	–	–

5) In initial step, while labeling of x_{-1} and x_0 , it is advisable to choose that point as x_0 which is nearer to the root. For it, $f(x_0)$ and $f(x_{-1})$ can be compared and the point with smaller function value may be x_0 and other x_{-1}.

6) It is advised to use $x_{n+1} = x_n - \dfrac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$ for computational reasons.

7) There are other methods which begin with three initial guesses of the root and instead of using straight line employ quadratic interpolating polynomial example: Brent's method, Muller's method. In case of Muller's method convergence is super linear, order of convergence is 1.84, higher than that of Secant Method.

8) Both initial guesses can be on one side of the root.



We

[Figure 5]

find, as observed, Secant Method is quite fast, it has converged very fast. It is very good economical method.

The only caution required in selection of proper initial guesses, so that the method does not diverge. For it, if the method is diverging, it implies for that particular

epathshala पाठशाला A Gateway to All Post Graduate Courses

equations, initial guesses are not sufficiently near the root. The best course of action would be to solve equation in two phases:

Phase – I:

Find estimate of the root using Bisection Method, correct to certain number of digits, which is always feasible as Bisection method though slow, is steady and reliable.

Phase – II:

The interval end points of last iteration of bisection method can be input to Secant method. This two phase hybrid method works very well, in the situations, when high accuracy results are needed in, to delivered with fast pace.

4.6 Comparison of speed of Bisection Method, Method of False Position, Secant Method

Example:

$f(x) = 3x - \cos x - 1$

$a = 0, f(a) = -2.000000$

$b = 0, f(b) = 1.459698$

k	Bisection Method		False Position Method		Secant Method	
	c_k	$f(c_k)$	c_k	$f(c_k)$	c_k	$f(c_k)$
1	0.500000	-0.377583	0.578085	-0.103255	0.578085	-0.103255
2	0.750000	0.518311	0.605959	-0.004081	0.605959	-0.004081
3	0.625000	0.064037	0.607057	-0.000159	0.607105	0.000014
4	0.562500	-0.158424	0.607100	-0.000006	0.607102	0.000000
5	0.593750	-0.047598	0.607102	-0.000000	–	–
6	0.609375	0.008119	–	–	–	–
7	.	.	–	–	–	–
8	.	.	–	–	–	–

k	Bisection Method		False Position Method		Secant Method	
	c_k	$f(c_k)$	c_k	$f(c_k)$	c_k	$f(c_k)$
9	•	•	–	–	–	–
19	0.607103	0.000006	–	–	–	–
20	0.607102	0.000000	–	–	–	–

4.7 Advantages and Disadvantages

Advantages

1) Open Method, no constraint of end points of interval to contain the root.
2) If converges, it converges quite fast, super linear convergence of order 1.618
3) Requires only one function evaluation per iteration.

Disadvantages

1) No more guarantee of convergence
2) May diverge, if initial guess are not chosen cautiously

Pathshala पाठशाला A Gateway to All Post Graduate Courses

4.8 Summary of characteristics of three methods, studied so far

	Bisection Method	False Position Method	Secant Method
1	Two initial guesses a, b with $f(a).f(b) < 0$	Two initial guesses a, b with $f(a).f(b) < 0$	Two initial guesses x_{-1}, x_0 No constraint.
2	Steady, Slow, Reliable	Faster than generally Bisection Method (Exceptions can be there)	Fast
3	Guarantees convergence	Guarantees convergence	Does not guarantee convergence
4	One function evaluation per iteration	One function evaluation per iteration	One function evaluation per iteration
5	Rate of converge $O\left(\frac{1}{2^n}\right)$ linear	Super linear	Super linear: 1.615

4.9 Speed of Convergence

We have used this terminology of rate of convergence and order of convergence in comparing the three methods, to find roots of an equation $f(x) = 0$, we have studied so far. It is right time to understand these concepts.

The rate of convergence and order of convergence are two principal measures of convergence speed of iterative processes.

Rate of Convergence

Let sequence $\{c_k\}$ converge to say *c*. if there exists a sequence $\{\beta_k\}$ which converges to 0 and a positive constant λ, independent of k, such that $|c_k - c| \le \lambda(\beta_k)$, $k \ge k_0$ then $\{c_k\}$ is said to converge to *c* with rate of convergence as $\mathrm{O}(\beta_k)$.

Order of Convergence

Let $\{c_k\}$ converge to *c*.

Let error $e_k = c_k - c$ for $k \ge 0$.

If there exists a positive constant λ and α such that

$$\lim_{k\to\infty}\frac{|c_{k+1} - c|}{|c_k - c|^{\alpha}} = \lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|^{\alpha}} = \lambda,$$

then $\{c_k\}$ is said to converge to *c* of ORDER α with asymptotic error constant λ.

$$|e_{k+1}| \approx |e_k|^{\alpha}$$

An iterative method is called of order α, If the sequence of approximations generated by it converges of order α.

If α =1, it is called linear convergence

α =2, it is called quadratic convergence

α =3 it is called cubic convergence

____________ x _____________