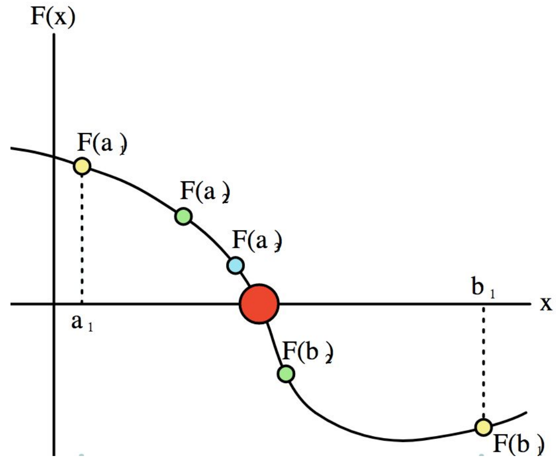
Programming Assignment 1
ESO 208

Name	Adarsh Srivastava
Roll Number	150047
Section	K-2

The program is written in **JavaScript**, the executable HTML file is named **Assignment.html** and the corresponding JavaScript file containing the code is named **numerical.js**

To run the program just **double click on the html file** and select the method you want to use, after entering all the required input by a method press the **Compute** button.

Bisection Method -



The *bisection method,* which is alternatively called binary chopping, interval halving, or Bolzano's method, is one type of incremental search method in which the interval is always divided in half.

Disadvantages -

1. Converges very slowly.
2. Values of function not taken into account while creating intervals.

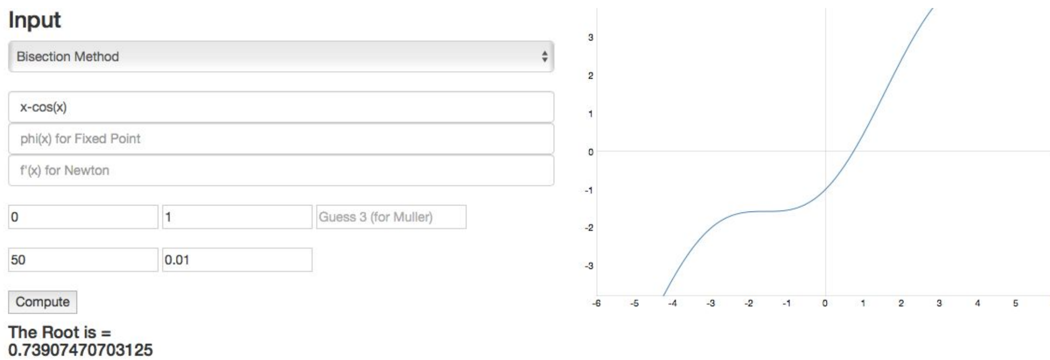
The bisection method fails to converge in the second case, as the relative closeness of root to 1 is not taken into account and therefore the bracketing fails.

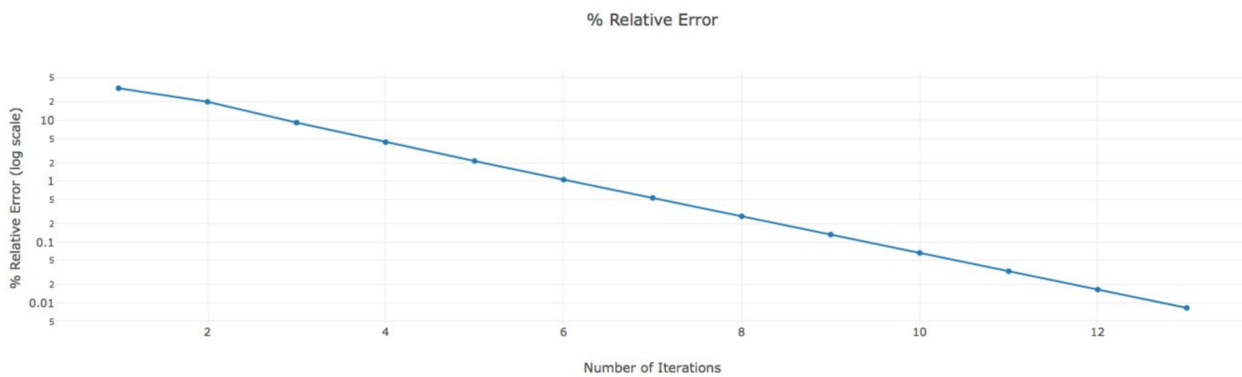
Convergence -

The method is guaranteed to converge to a root of f if f is a continuous function on the interval $[a, b]$ and $f(a)$ and $f(b)$ have opposite signs. The absolute error is halved at each step so the method converges linearly, which is comparatively slow.

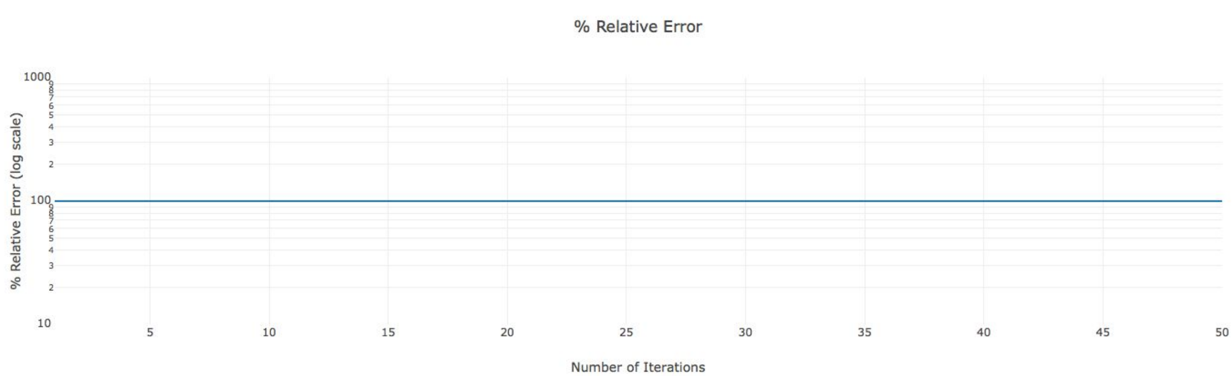
Specifically, if $c_1 = a+b/2$ is the midpoint of the initial interval, and c_n is the midpoint of the interval in the nth step, then the difference between c_n and a solution c is bounded by

Test Case 1 -

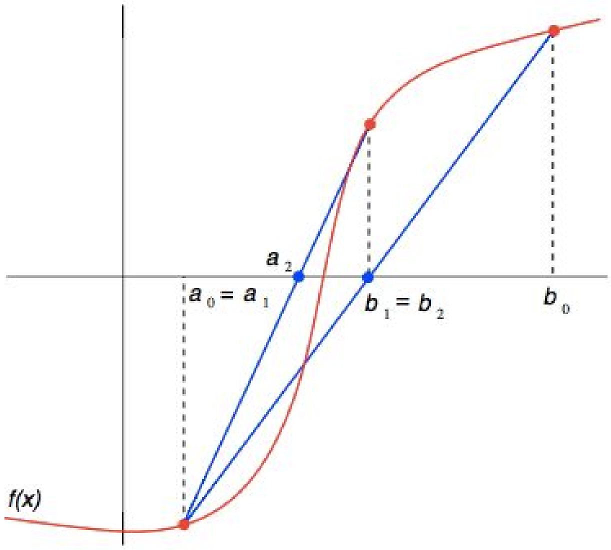




Test Case 2 -



False Position Method -

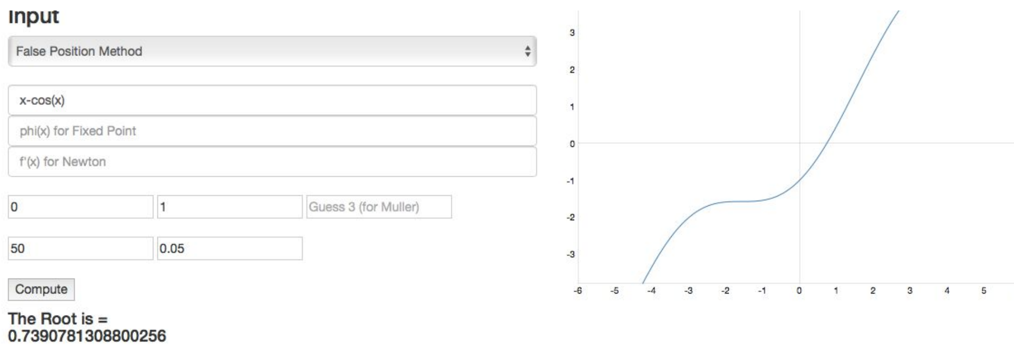


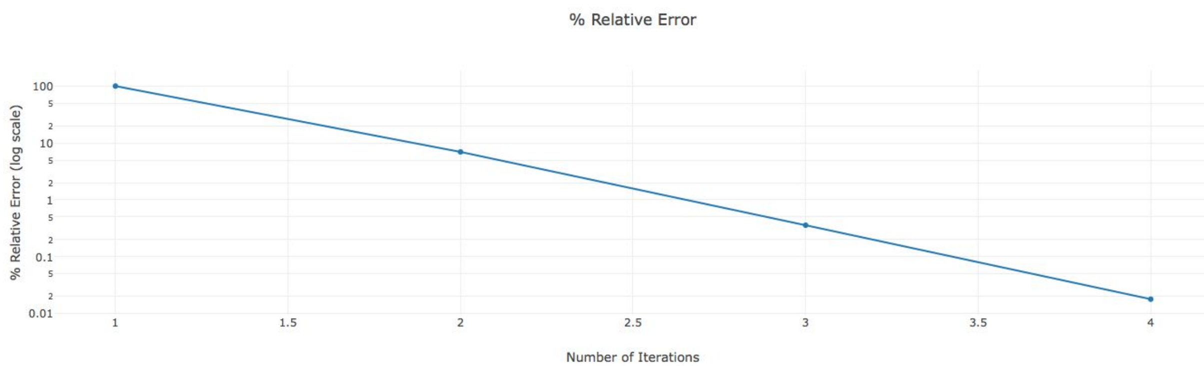
An alternative method that exploits this graphical insight is to join f(xl) and f(xu) by a straight line. The intersection of this line with the x axis represents an improved estimate of the root. The fact that the replacement of the curve by a straight line gives a “false position” of the root is the origin of the name, method of false position, or in Latin, regula falsi. It is also called the linear interpolation method.

Convergence -

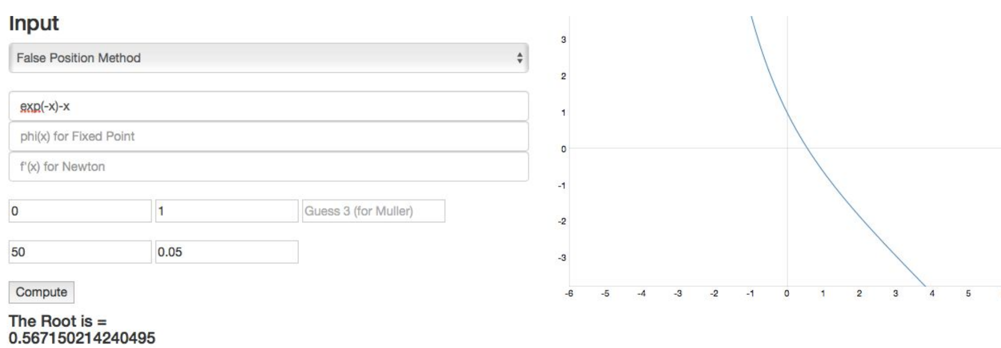
If the initial end-points a_0 and b_0 are chosen such that $f(a_0)$ and $f(b_0)$ are of opposite signs, then at each step, one of the end-points will get closer to a root of *f*. If the second derivative of *f* is of constant sign (so there is no inflection point) in the interval, then one endpoint (the one where *f* also has the same sign) will remain fixed for all subsequent iterations while the converging endpoint becomes updated. As a result, unlike the bisection method, the width of the bracket does not tend to zero (unless the zero is at an inflection point around which *sign(f)=-sign(f")*). As a consequence, the linear approximation to $f(x)$, which is used to pick the false position, does not improve in its quality.

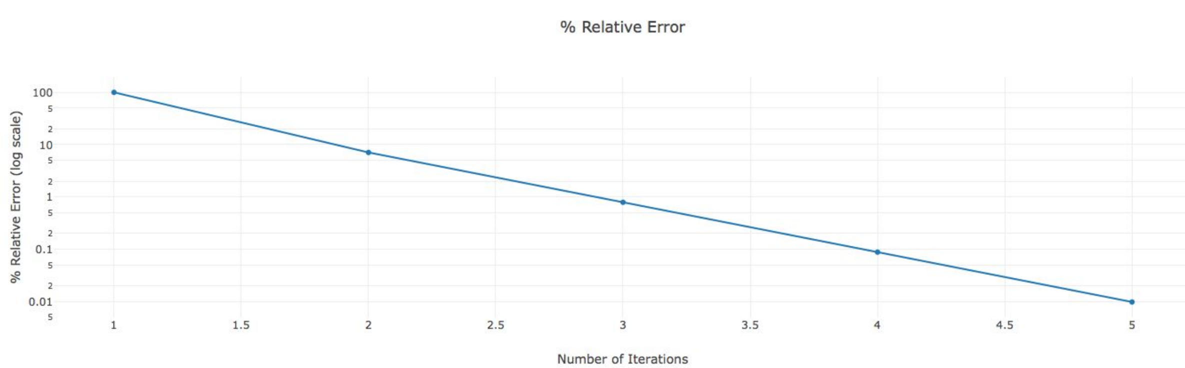
Test Case 1 -



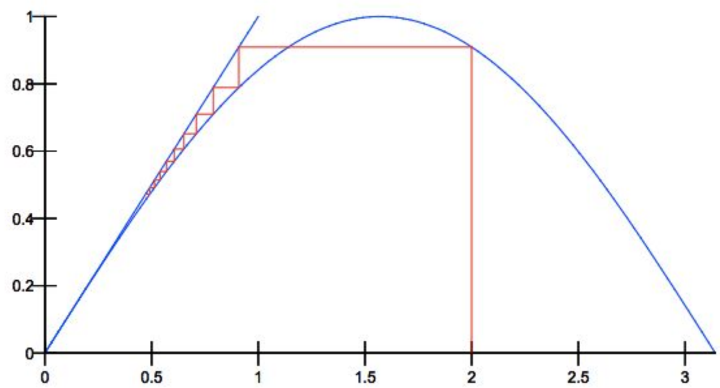


Test Case 2 -





Fixed Point Method -



Fixed-point iteration is a method of computing fixed points of iterated functions.

More specifically, given a function f defined on the real numbers with real values and given a point x0 in the domain of f, the fixed point iteration is

$x_{n+1}=f(x_n), n=0,1,2,\dots$

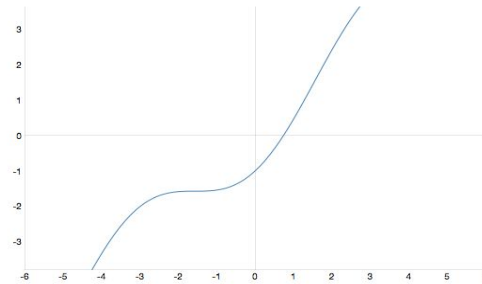
which gives rise to the sequence $x_0, x_1, x_2, \dots$ which is hoped to converge to a point x. If f is continuous, then one can prove that the obtained x is a fixed point of f, i.e.,

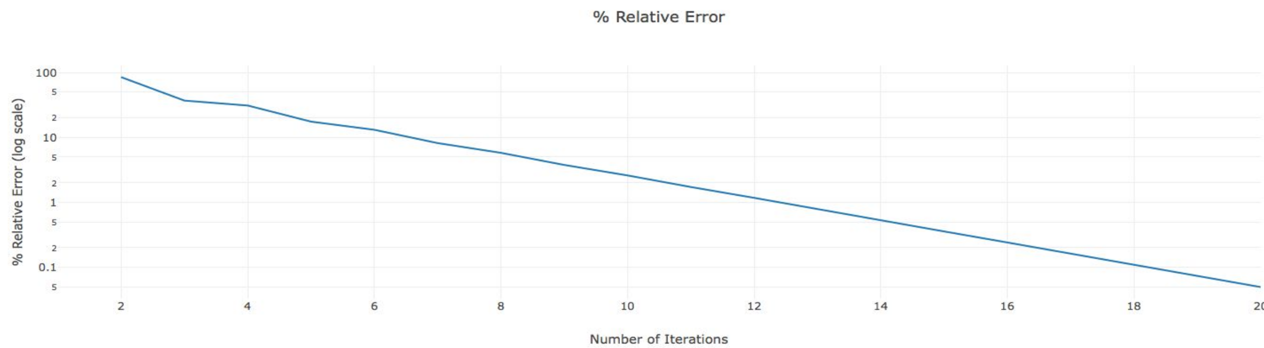
$f(x)=x.$

Convergence -

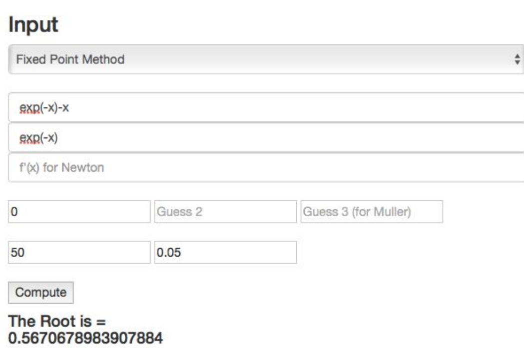
Linear Convergence, is characteristic of fixed-point iteration.

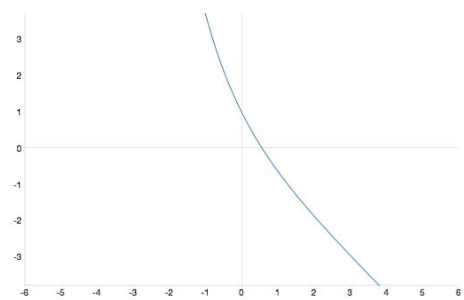
Test Case 1 -

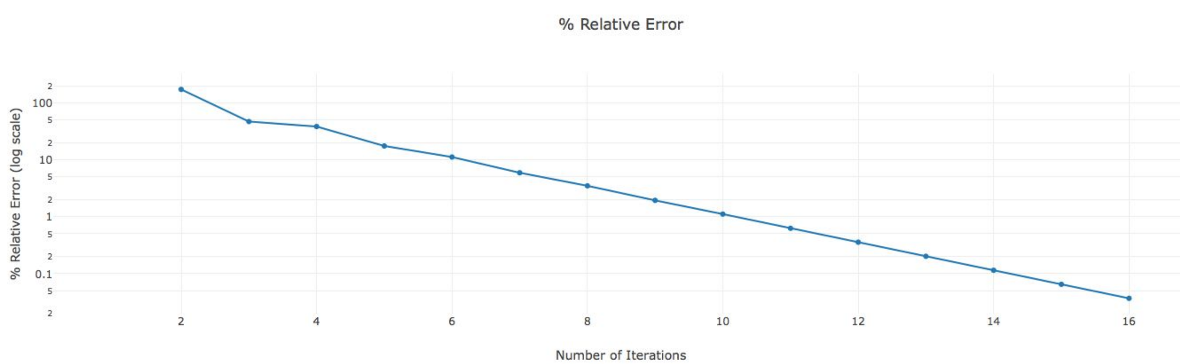




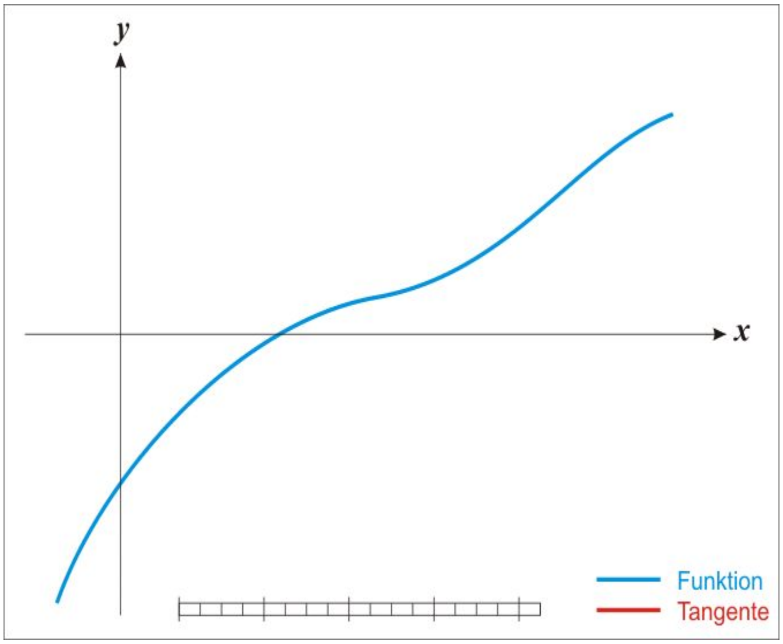
Test case 2 -







Netwon Method -



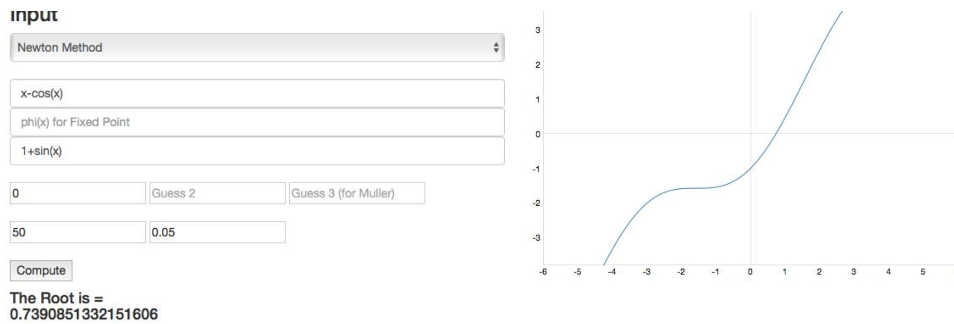
The method starts with a function f defined over the real numbers x, the function's derivative $f\ '$, and an initial guess x_0 for a root of the function f. If the function satisfies the assumptions made in the derivation of the formula and the initial guess is close, then a better approximation x_1 is

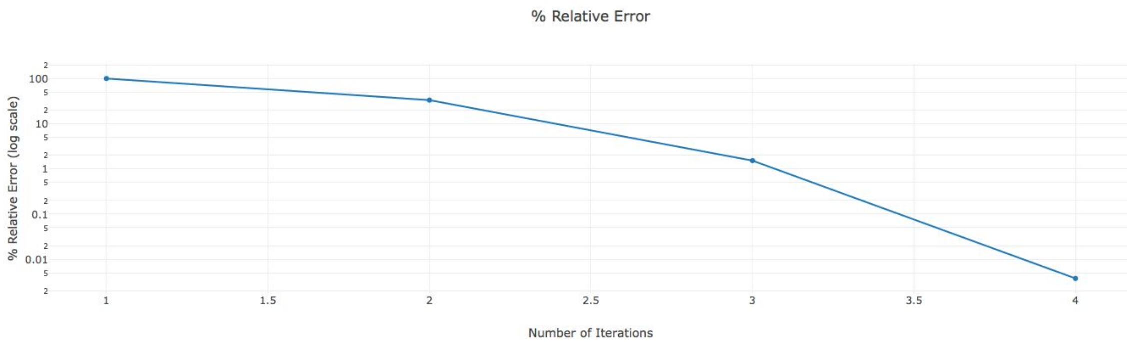
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Convergence -

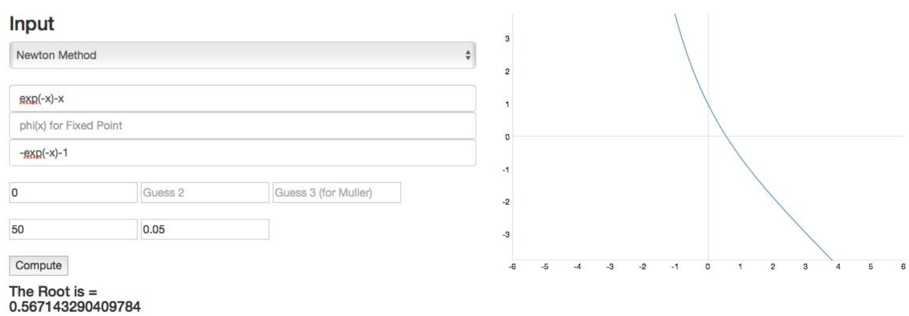
If the root being sought has multiplicity greater than one, the convergence rate is merely linear (errors reduced by a constant factor at each step) unless special steps are taken. When there are two or more roots that are close together then it may take many iterations before the iterates get close enough to one of them for the quadratic convergence to be apparent. However, if the multiplicity *m* of the root is known, the following modified algorithm preserves the quadratic convergence rate

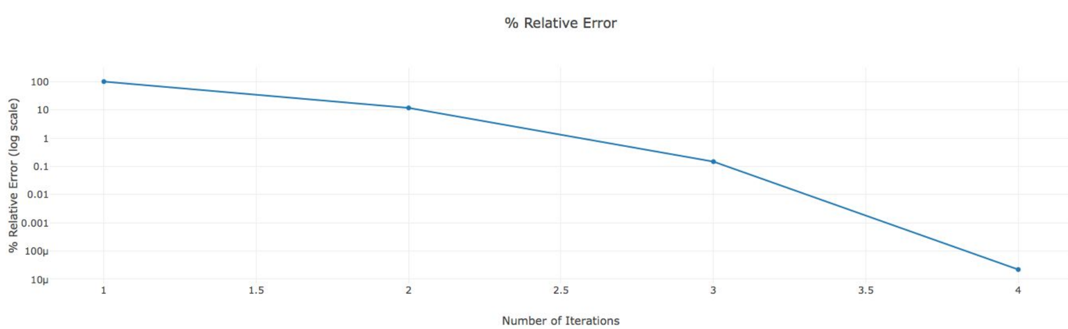
Test Case 1 -



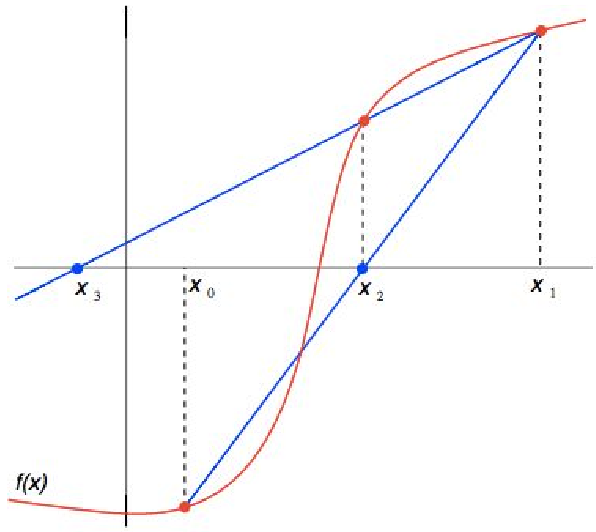


Test Case 2 -





Secant Method -

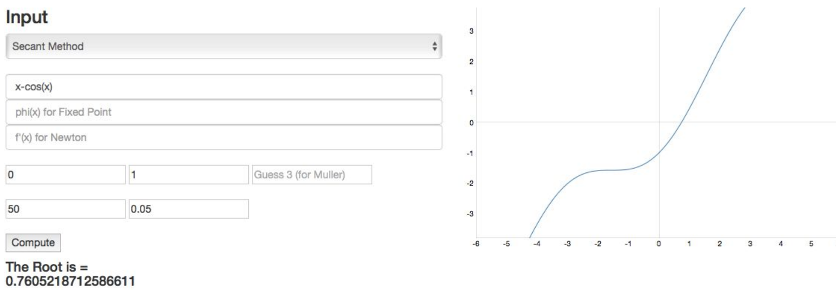


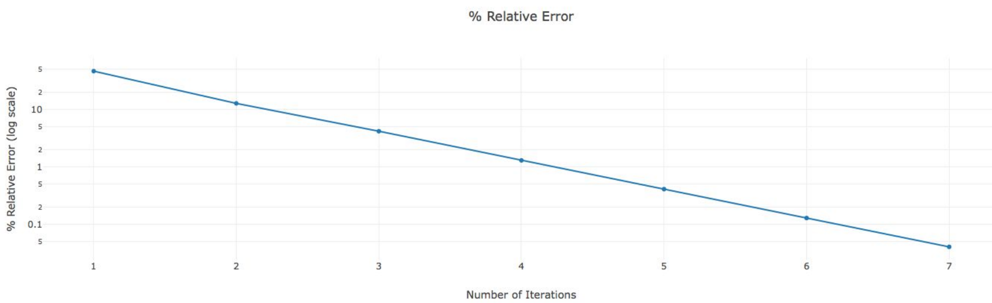
Convergence -

The iterates *xn* of the secant method converge to a root of *f,* if the initial values *x0* and *x1* are sufficiently close to the root. The order of convergence is *α*, where it is the golden ratio. In particular, the convergence is superlinear, but not quite quadratic.

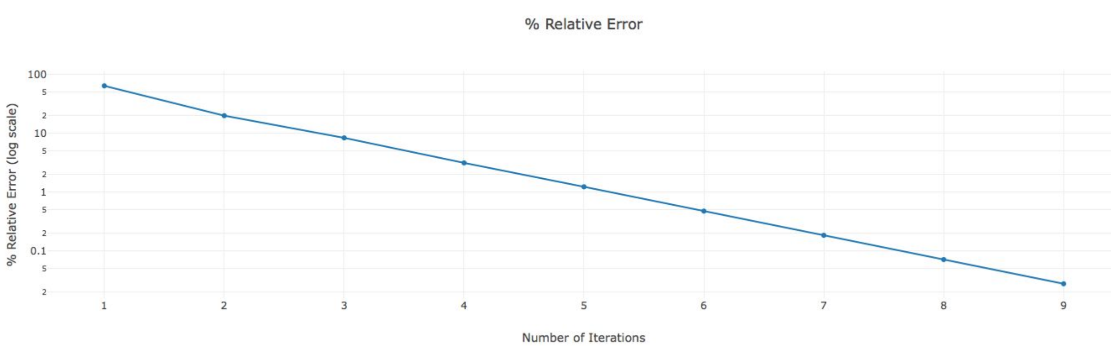
f the initial values are not close enough to the root, then there is no guarantee that the secant method converges. There is no general definition of "close enough", but the criterion has to do with how "wiggly" the function is on the interval *[x0, x1]*. For example, if f is differentiable on that interval and there is a point where *f′=0* on the interval, then the algorithm may not converge.

Test Case 1 -





Test Case 2 -



Muller Method -

Muller's method is based on the secant method, which constructs at every iteration a line through two points on the graph of *f*. Instead, Muller's method uses three points, constructs the parabola through these three points, and takes the intersection of the *x*-axis with the parabola to be the next approximation.

Convergence -

The order of convergence of Muller's method is approximately 1.84. This can be compared with 1.62 for the secant method and 2 for Newton's method. So, the secant method makes less progress per iteration than Muller's method and Newton's method makes more progress.

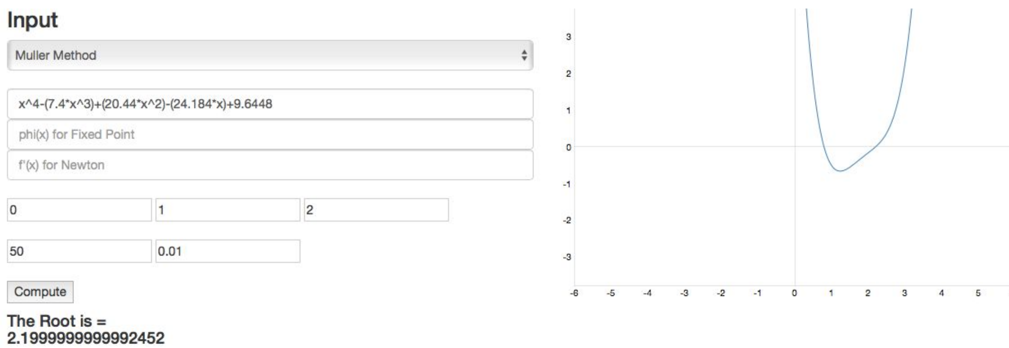
$$\lim_{k\to\infty} \frac{|x_k - \xi|}{|x_{k-1} - \xi|^{\mu}} = \left| \frac{f'''(\xi)}{6f'(\xi)} \right|^{(\mu-1)/2},$$

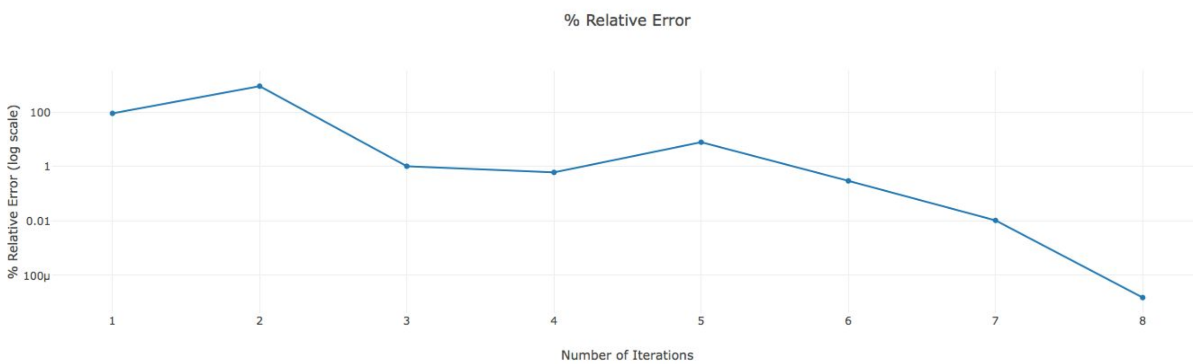
where $\mu \approx 1.84$

Note -

We do not use the standard formula for solving quadratic equations because that may lead to loss of significance.

Test Case 1 -





Test Case 2 -