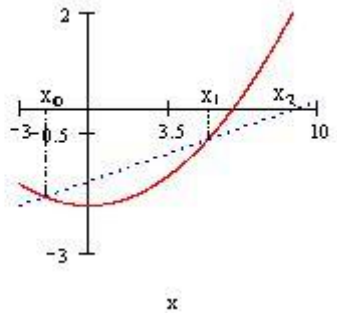
SECANT METHOD

The Newton-Raphson algorithm requires the evaluation of two functions (the function and its derivative) per each iteration. If they are complicated expressions it will take considerable amount of effort to do hand calculations or large amount of CPU time for machine calculations. Hence it is desirable to have a method that converges (please see the section order of the numerical methods for theoretical details) as fast as Newton's method yet involves only the evaluation of the function. Let x_0 and x_1 are two initial approximations for the root 's' of $f(x) = 0$ and $f(x_0)$ & $f(x_1)$ respectively, are their function values. If x_2 is the point of intersection of x-axis and the line-joining the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ then x_2 is closer to 's' than x_0 and x_1. The equation relating x_0, x_1 and x_2 is found by considering the slope 'm'



$$m = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{0 - f(x_1)}{x_2 - x_1}$$

$$x_2 - x_1 = \frac{-f(x_1) * (x_1 - x_0)}{f(x_1) - f(x_0)}$$

$$x_2 = x_1 - \frac{f(x_1) * (x_1 - x_0)}{f(x_1) - f(x_0)}$$

or in general the iterative process can be written as

$$x_{i+1} = x_i - \frac{f(x_i) * (x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \qquad i = 1,2,3...$$

This formula is similar to Regula-falsi scheme of root bracketing methods but differs in the implementation. The Regula-falsi method begins with the two initial approximations 'a' and 'b' such that $a \leq s \leq b$ where s is the root of $f(x) = 0$. It proceeds to the next iteration by calculating $c(x_2)$ using the above formula and then chooses one of the interval (a,c) or (c,h) depending on $f(a) * f(c) < 0$ or > 0 respectively. On the other hand secant method starts with two initial approximation x_0 and x_1 (they may not bracket the root) and then calculates the x_2 by the same formula as in Regula-falsi method but proceeds to the next iteration without bothering about any root bracketing.

Algorithm - Secant Method

Given an equation f(x) = 0
Let the initial guesses be x_0 and x_1
Do

$$x_{i+1} = x_i - \frac{f(x_i) * (x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \qquad i = 1,2,3...$$

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 $| x_{i+1} - x_i |$ (where i is the iteration number) less than some tolerance limit, say epsilon, fixed apriori.

Numerical Example :

Find the root of **3x+sin[x]-exp[x]=0** [*Graph*]

Let the initial guess be **0.0** and **1.0**

f(x) = 3x+sin[x]-exp[x]

i	0	1	2	3	4	5	6
x_i	0	1	0.471	0.308	0.363	0.36	0.36

So the iterative process converges to 0.36 in six iterations.

Worked out problems

Exapmple 1	Find a root of cos(x) - x * exp(x) = 0	**Solution**
Exapmple 2	Find a root of x^4-x-10 = 0	**Solution**
Exapmple 3	Find a root of x-exp(-x) = 0	**Solution**
Exapmple 4	Find a root of exp(-x) * (x^2-5x+2) + 1= 0	**Solution**
Exapmple 5	Find a root of x-sin(x)-(1/2)= 0	**Solution**
Exapmple 6	Find a root of exp(-x) = 3log(x)	**Solution**
Problems to workout		

Work out with the **SECANT** method here

Note : Few examples of how to enter equations are given below . . . (i) exp[-x]*(x^2+5x+2)+1 (ii) x^4-x-10 (iii) x-sin[x]-(1/2) (iv) exp[(-x+2-1-2+1)]*(x^2+5x+2)+1 (v) (x+10) ^ (1/4)