Design and Analysis of Algorithm for finding roots of a quadratic equation using Secant Method

Nairit Banerjee
IIT2016505

Kamal Nayan Chaubey
IIT2016510

Ashutosh Vishwakarma
IIT2016017

Mandeep Chakma
IIT2016012

February 21, 2018

Abstract

This document is a final report of our study on finding the roots of a quadratic equation of the form $x^2 - n$ using Secant Method.

Keywords: *Interval,function,error,Accuracy, Number of iterations*

1 Introduction And Literature Survey

In mathematics and computer science, the process of finding the roots of a equation has number of methods. One such method is Secant Method. A method that takes interval,error as an input and outputs the approximate roots of that equation.
Our equation is of the form f(x) = x^2−n, where n will be taken as input.For example we have an equation f(x) = x^2 - 4,by simple method we know that root of the above equation is 2,-2 as f(2) = f(-2) = 0.
But secant method is as follows :-
Assume an interval **[a,b]** where we know that one of the root of equation lies in the given interval i.e $c \in [a, b]$ such that **f(c) = 0** and f(a)*f(b) < 0. c is calculated as follows:

$$c = \frac{(a * f(b) - b * f(a))}{(f(b) - f(a))} ....(1)$$

The above formula is calculated by using the basic definition of slope of a curve at a point.

$$f'(c) = \frac{(f(b) - f(a))}{b - a}$$

, where

$$f'(c) = \frac{f(b)}{b - c}$$

, on solving for the value of c we get the eq(1).

2 Algorithm Design

The first task in Algorithm design to be done is to produce the interval where our root will lie. The algorithm description for the problem is as follows:

(1) procedure Secant method
We start with two estimates of the root x_0 and x_1,where x_0 being the value of x where the value of the function $x^2 - n$ is less than 0 and x_1 be the value of x where the function is greater than zero.So clearly root must lie in between these two points.Now we recursively find the value x_{n+1} by use of x_{n-1} and x_n by using the following formula for $n \geq 1$:

$x_{n+1} = x_n - f(x_n)/Q(x_{n-1}, x_n)$,
Where ,
$Q(x_{n-1}, x_n) = (f(x_{n-1}) - f(x_n))/(x_{n-1} - x_n)$.
Now as the value of x_{n+1} comes we check the value of $f(x_{n+1})$.If it is 0 then we simply print the roots else we iterate till the difference $x_{n+1} - x_n \geq e$ where e is the degree of accuracy and is taken equal to 10^-6 and then the final value of x_{n+1} that comes after the loop finishes, is printed as the root of quadratic equation which is very close to the actual root.

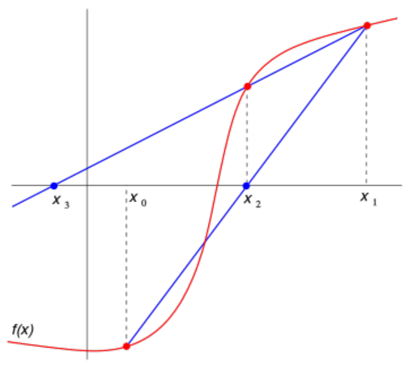

Figure 1: Secant Method diagram.

2.1 Pseudo Code

Algorithm 1 Function Value

```
procedure F($x$)
    $a \leftarrow x^2 - n$
    return $a$
end procedure
procedure Main
    if $n == 0$ then
        Roots are both $= 0$
        return 0
    end if
    if $n < 0$ then
        Roots are imaginory
        return 0
    end if
    while $F(i) \leq 0$ do
        $i \leftarrow i + 1.0$
    end while
    $b \leftarrow i - 1$
    $c \leftarrow i$
    $e \leftarrow 10^{-6}$
    while $abs(c - b) \geq e$ do
        $a \leftarrow b$
        $b \leftarrow c$
        $c \leftarrow b-((b-a)/(F(b)-F(a)))*F(b)$
        if $F(c) == 0$ then
            The first root of equation is c
            The second root of equation is -c
            return 0
        end if
    end while
    The first root of equation is c
    The second root of equation is -c
    return 0
end procedure
```

3 Analysis and Discussions

3.1 Time Complexity

The time complexity for the above stated algorithm can be computed by adding units of time taken by each statement multiplied by the number of times the statement is executed. Hence,

$$T = \Sigma(cost(i) * no.of.times(i)) \text{ ,for all statements}$$

The running time will be dependent on the coefficient n in the equation

$$x^2 - n = 0$$

If n is zero or negative, it results in best case and returns output in constant time. Whereas if n is any positive number, the algorithm will take time to estimate the range between which a root may lie. In other words time taken will be dependent on how far the control has to travel along the x-axis to find a change in nature(sign) of the function.

$$T_{best} \propto c_1$$
$$T_{average} \propto \sqrt{n} + 10c_2$$
$$T_{worst} \propto \sqrt{n} + 10c_3$$

Hence the order of growth for best case, average case and worst case are

$$\Omega(1), \Theta(\sqrt{n}), O(\sqrt{n}) \text{ respectively.}$$

The average and worst case differ in constant units of time. This is because if n is a perfect square, one of the ends of the estimated range will be a root. Whereas if n is not a perfect square , we now need to actually iterate through the secant method to reach to the root.

For the dataset,

Table 1: Data for plotting running time graph

n	T_{best}	T_{avg}	T_{worst}
0	4	-	-
-100	5	-	-
64	-	83	-
625	-	163	-
69	-	-	385
1001	-	-	249

We will get the following time complexity graph for the above stated and a few more values of n

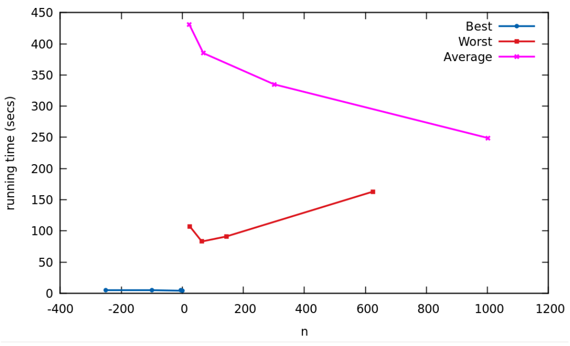

Figure 2: Running time graph.

Now we will plot a graph between the number of iterations for which we run the secant function and the error percentage in finding the root of the equation

$$x^2 - 67 = 0$$

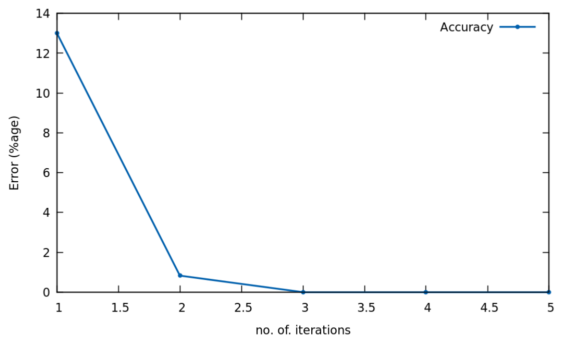

Figure 3: Percentage Error Graph

3.2 Space Complexity

The algorithm does not consume any extra space in the memory other than the variables used. Hence space complexity is linear.

$$Space \propto 1$$

4 Experimental Study

We may assume the following examples to study the various running times possible,

Best Case: The best case will depend on the value of n :
(i)when **n=0**, in this both roots will be **0**.
(ii)when **n<0**, in this case **imaginary roots**

Average Case: In this case the input n will be a perfect square number.So the number of iteration in this case will be lower than for the number which is not a perfect square.Infact in only one iteration we get the root in this case.For example suppose the value of n is 64 then it tales only one iteration to get to the root.

Worst Case:In this case the input n will be any random number.So the number of iteration in this case may be greater than or equal to one.For example say n = 63 then it takes 4 iterations to get the root.

5 Conclusion

In this paper we proposed an algorithm to find the roots of a quadratic equation of the form :
x^2 - n using **Secant method**.

The Secant Method is an iterative method in which the peak displacement response of a structure or structural component is determined from linear dynamic analyses of a model whose stiffness is updated to reflect a computed degree of degradation that is consistent with the computed peak displacement.The Secant Method is one of a number of analytical procedures available to earthquake engineers today for predicting the earthquake performance of structures.

References

[1] Introduction To Algorithms (Second Edition) by Thomas H.Cormen, Charles E.Leiserson , R.L.Rivest and Clifford Stein.

[2] How to Solve it by Computer by R.G.Dromey.

[3] The C Programming Language by Brian Kernighan and Dennis Ritchie.

[4] https://www.geeksforgeeks.org/

[5] https://www.stackoverflow.com/

[6] http://mathworld.wolfram.com/SecantMethod.html