TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS
Dharan-8

A Lab Report On: To Find A Root Of Non-Linear Using Secant Method

Submitted By
Name: Dhiraj KC
Roll No. : PUR077BEI014
Faculty: Electronics, Communication and Information
Group: A

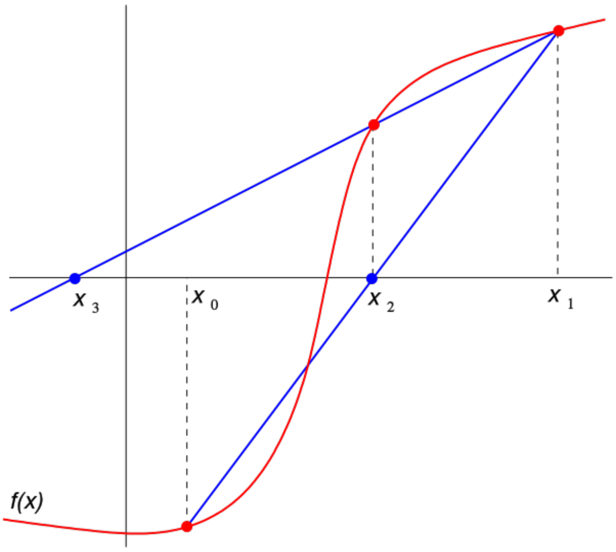
Submitted To
Department Of Electronics and Computer Engineering
Checked By: Pukar Karki

TITLE: TO FIND A ROOT OF NON-LINEAR USING SECANT METHOD

THEORY

Secant method is also a recursive method for finding the root for the polynomials by successive approximation. It's similar to the Regular-falsi method but here we don't need to check $f(x_1)f(x_2) < 0$ again and again after every approximation. In this method, the neighbour roots are approximated by secant line or chord to the function $f(x)$. It's also advantageous of this method that we don't need to differentiate the given function $f(x)$, as we do in Newton-Raphson method.

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



ALGORITHM

1. Start
2. Define function $f(x)$
3. Initialize Different Constants
4. Using the formula above to calculate the value in every iteration.
5. Display
6. Stop

PROGRAM

```
import math

def fun(x):
  return x*x*x-18

def secantCalcuator(a, b, err):
  step = 1
  x0 = a
  x1 = b
  f0 = fun(a)
  f1 = fun(b)
  flag = True
  if (f0*f1 > 0):
    print("The solution between these point is not converging")
  while flag:
    x2 = (f1*x0 - f0*x1)/(f1-f0)
    f2 = fun(x2)
    print('Iteration - %d, x2 = %0.8f and f(x2) = %0.8f' % (step, x2, f2))
    # now again checking whether the solution is converging or not
    if (f0*f2 < 0):
      x1 = x2
    else:
      x0 = x2
    step += 1
    flag = abs(f2) > err
  print("The root is %0.6f", x2)

secantCalcuator(2, 3, 0.001)
```

OUTPUT

```
Iteration - 1, x2 = 2.52631579 and f(x2) = -1.87636682
Iteration - 2, x2 = 2.77562327 and f(x2) = 3.38363628
Iteration - 3, x2 = 2.65753025 and f(x2) = 0.76871982
Iteration - 4, x2 = 2.59537603 and f(x2) = -0.51760738
Iteration - 5, x2 = 2.62808878 and f(x2) = 0.15181666
Iteration - 6, x2 = 2.61259327 and f(x2) = -0.16736952
Iteration - 7, x2 = 2.62074880 and f(x2) = 0.00015261
The root is %0.6f 2.620748800740416
```

DISCUSSION

From this practical, it can be concluded that the secant method is faster than bisection method in terms of converging and the result was found to be 2.62074.