KITAVI DUNCAN GITAU

#SCT211-0031/2021

INTRODUCTION

It is an iterative numerical technique used for finding the roots of a real-valued function.

By using the idea that a continuous and differentiable function can be approximated by a straight line tangent to it.

The method starts by first guessing the root. root of the function and iteratively refines this guess until it converges to the actual root.

One initializes by guessing x with its func $f(x)$

After the guess one computes the x_n+1 using formulae: $x_n+1=x_n-\frac{f(x_n)}{f'(x_n)}$

where $f'(x_n)$ is the derivative.

After converging in reading tolerance level.

x_n+1 is the estimated root of the func.

Significance Of the Newton-Raphson method in numerical Analysis

-Used as a technique for finding the roots of given equations.

-For efficiently generalised to find solutions to a system of equations.

-Can doquadratic convergence near a simple root.

-Use for optimization of algorithms, such as Newton's method for optimization

THEORY

. FORMULAE -

A value $ x = f(x) = 0$ The Newton Raphson refines x_0 to get:

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

where:: $x_n + 1 $ = approx of the root.

x_n = current root approx.

$f(x_n)$ = value of current approx.

$f'(x_n)$ is the derivative.

Convergence criteria

It cconverges $|f(x).f''(x)|<|f'(x)|^2$ This method fails if $f'(x)=0$

This depends on the initial guess and the properties of the function $f(x)$.

To convert root: $f(x)$ should be close to actual root and close to the continous and differentiable in root.

The $f(x)$ should not be zero near the desired root.

Potential Limitations

- The necessity to start the iteration rather close to the searched-for zero in order to achieve convergence
- To calculate the inverse of the derivative at each iterative step one must ensure iteration is rather close to searched for zero.
- For functions with multiple roots, discontinuities, or oscillatory behavior, its is complex.
- computationally expensive to compute due to multiple roots oscillatory behavior.

```
import numpy as np

#this is functionf(x) = x^2 - 4
def func(x):
  return x**2 - 4


#this is its derivative f'
#which is  2 * x
def derivedfunc(x):
  return 2 * x


#Func to find the root
def Raphson(x):
  h = func(x) / derivedfunc(x)

  while abs(h) >= 0.0001: #h, is less than 0.0001, indicating
convergence.
    h = func(x) / derivedfunc(x)

    x = x -h

    print("The root is: ","%.4f"% x)
```

```
x0 = -70  #assumption of the initial values
Raphson(x0)
```

```
The root is:  -35.0286
The root is:  -17.5714
The root is:  -8.8995
The root is:  -4.6745
The root is:  -2.7651
The root is:  -2.1059
The root is:  -2.0027
The root is:  -2.0000
The root is:  -2.0000
```

Roots Found Above are:

The root is: -35.0286

The root is: -17.5714

The root is: -8.8995

The root is: -4.6745

The root is: -2.7651

The root is: -2.1059

The root is: -2.0027

The root is: -2.0000

The root is: -2.0000

Efficiency

-It adequately converges the root with a quadratic convergence rate, meaning that the number of correct digits roughly doubles with each iteration.

-It converges to required accuracy.

-It can be efficiently generalised to find solutions to a system of equations.

Accuracy

-An error tolerance of ϵ = 0.0001 is used. -Depending on the correctness of the derivative function the accuracy is determined.

Convergence

-The order of convergence of Newton Raphson method is 2.

-Convergence depends on the properties of the function and the initial guess.

f and f' exist and are continuous, and $f'(x)$ = 0, and 3. x_0 is close to x'.

Challenges

-Difficulty in calculating the derivative of a function.

-Slow convergence for roots of multiplicity greater than 1.

-Ensuring convergence to the desired root is a challenge.

-Accuracy of the derivative function. If the derivative is not computed correctly, it can lead to inaccurate results or convergence issues.

Summary

In summary Newton Raphsonan is an iterative numerical method used to find the roots of a real-valued function.

with the formula: $x_n+1=x_n-\frac{f(x_n)}{f'(x_n)}$

From the code above the Newton Raphson implements its arithmetic to get roots from its formulae through convergence and derivatives.

It is a way to quickly find a good approximation for the root of a real-valued function $f(x)=0 f(x)=0 f(x)=0$

Applicability

- In Machine Learning by using algorithms to iterate through functions procedual.
- In Mathematics in solution calculation ,problems, numerical optimization.
- In Numerical Analysis, used for finding roots of nonlinear equations arising from mathematical models and simulations.
- In Biology solutions in biological networks.
- Utilized for solving partial differential equations (PDEs) and nonlinear equations.
- In Engineering for structural analysis.
- In Physics to apply nonlinear eigenValue.

References

1.Luciano Ramal:Data with Numpy projects,Google books.

2.VanderPlas, J. (2016). Python Data Science Handbook:

3.W3Schools on python implementation