JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

BSc. COMPUTER SCIENCE

SCIENTIFIC COMPUTING - ICS 2207

DANIEL MBUGUA KARUME

SCT211-0072/2022

APPLICATION OF NEWTON-RAPHSON METHOD IN FINDING ROOTS OF EQUATIONS

Monday, Mar 3, 2024

INTRODUCTION

The Newton-Raphson method, a powerful iterative numerical technique for finding roots of real-valued functions, was independently discovered by Sir Isaac Newton and Joseph Raphson in the 17th century. Newton initially developed the method in the context of solving problems in calculus and analysis, while Raphson, a mathematician and merchant, later published the technique in his 1690 book "Analysis Aequationum Universalis."

The method's formulation involves using iterative approximations based on the function's value and its derivative, demonstrating the convergence towards the root.

This method is used to find the roots of a real-valued function. It is particularly effective for finding the roots of nonlinear equations and is widely employed in numerical analysis and optimization.

THEORY

The Newton-Raphson method is rooted in calculus, leveraging Taylor's series expansion to locally approximate a real-valued function. This iterative approach utilizes the function's value and its derivative at each step, converging towards the root based on linear approximations.

The method's mathematical foundation relies on the continuity and differentiability of the involved derivatives, ensuring its efficacy in approximating the roots for various functions.

Formula

The method utilizes the following iterative formula:

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

Where:

- x_{n+1} is the next approximation of the root.
- x_n is the current approximation.
- $f(x_n)$ is the function value at x_n.
- $f'(x_n)$ is the derivative of the function at x_n.

Convergence Criteria

The method converges when $|f(x_n)|$ is sufficiently close to zero, indicating proximity to a root. Typically, convergence is achieved when $|f(x_n)|$ falls below a predefined tolerance or when the change in x becomes negligible.

This can be represented Mathematically using the formula:

$$|f(x) \cdot f''(x)| < |f'(x)|^2$$

Advantages

1. **Efficiency:**
   - Newton-Raphson often converges rapidly, requiring fewer iterations.
2. **Versatility:**
   - It applies to a wide range of functions and equations.
3. **Simplicity:**
   - The formulation of the Newton-Raphson method is straightforward, making it easy to understand and apply.

Limitations

1. **Sensitivity to Initial Guess:**
   - Convergence depends on a reasonable initial guess.
2. **Limited Applicability:**
   - May fail or diverge for certain functions or near singular points.
3. **Can only be used as an approximation method:**
   - Newton-Raphson method provides an iterative approach to find solutions, and while it often converges quickly, it's important to note that it provides approximations rather than exact solutions.
4. **Real Function Constraint:**

   - The Newton-Raphson method is suitable for real-valued functions, meaning that the function's output must be a real number for any real input.
   - The method relies on the evaluation of the function and its derivative, so the domain and range of the function should be real numbers.
5. **Continuous Derivative Constraint:**

   - The function's derivative, denoted as $f'(x)$, must be continuous in the interval of interest.

- The continuity of the derivative ensures smooth and predictable behavior, allowing the Newton-Raphson method to iteratively converge to the root effectively.
- Discontinuities or abrupt changes in the derivative may affect the convergence or lead to divergence of the method.

IMPLEMENTATION

In this section, the algorithm used in solving equations using the Newton-Raphson method is discussed and later implemented in as a Python program.

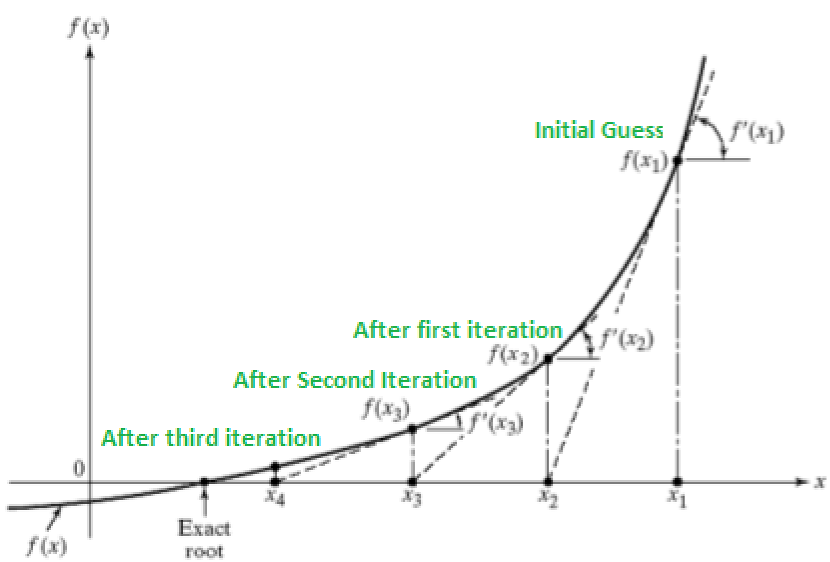
Algorithm

The steps that are used to solve an equation using the Newton-Raphson method are as follows:

1. **Initialize:** Choose an initial guess x_0.
2. **Iterate:** For each iteration n:
   - Compute the function value $f(x_n)$ and its derivative $f'(x_n)$.
   - Update the guess using the formula: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
   - Check for convergence: If the absolute value of $f(x_{n+1})$ is below a specified tolerance or a maximum number of iterations is reached, exit the loop, that is: $|f(x) \cdot f''(x)| < |f'(x)|^2$
3. **Output:** The final value of x_{n+1} is the approximate root.

This algorithm is used for finding the root of a real-valued function $f(x) = 0$.

The above steps can be represented diagramatically in the graph plotted below:



Sample Equation

Here, we will solve $f(x) = x^2 - 25$ using The Completing Square Method to get the exact. roots then solve it again using The Newton-Raphson Method. These two answers will be used for comparison and analysis in the next section.

Solving $f(x) = x^2 - 25$ using The Completing Square Method

1. **Start with the quadratic equation:**

   $f(x) = x^2 - 25.$

2. **Identify the coefficient of x^2:**

   The coefficient is 1.

3. **Rewrite the quadratic equation by completing the square:**

   $f(x) = x^2 - 25$

   $= (x^2 - 0x) - 25$

   $= (x^2 - 0x + 0^2) - 25 - 0^2$

   $= (x - 0)^2 - 25$

Now, the quadratic equation is in the form $(x - h)^2 - k$, where h is the x-coordinate of the vertex, and k is the y-coordinate of the vertex. In this case, $h = 0$ and $k = 25$.

So, $f(x) = (x - 0)^2 - 25.$

Now, we can find the roots by setting $f(x)$ to zero and solving for x: $0 = (x - 0)^2 - 25$

$25 = (x - 0)^2$

$x - 0 = \pm\sqrt{25}$

$x = \pm 5$

Therefore, the roots of the equation $f(x) = x^2 - 25$ are $x = 5$ and $x = -5$.

Solving $f(x) = x^2 - 25$ using The Newton-Raphson Method (with LaTex)

1. The Newton-Raphson method uses the iterative formula:

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

where x_{n+1} is the new guess for the root, $f'(x_n)$ is the derivative of $f(x)$ evaluated at x_n, and x_n is the current guess.

1. Step 1: Calculate the derivative of $f(x)$:

   $f'(x) = 2x$

2. Step 2: Choose an initial guess for the root, x_0.Let's take $x_0 = 7$.

3. Step 3: Plug x_0 into the formula to calculate x_1:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 7 - \frac{7^2 - 25}{2 \cdot 7} = 5.0714$$

4. Step 4: Repeat the process until the desired level of accuracy is achieved. Let's perform another iteration:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 5.0714 - \frac{(5.0714)^2 - 25}{2 \cdot 5.0714} = 5$$

5. After the second iteration, we have obtained an approximate root of $x = 5$.

6. To find the second root, consider a different initial guess.

7. Repeat the iterations:

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 5 - \frac{5^2 - 25}{2 \cdot 5} = -5$$

After the third iteration, we have obtained another approximate root of $x = -5$.

Therefore, using the Newton-Raphson method with an initial guess of $x_0 = 7$, we have found the roots $x = 5$ and $x = -5$.

Solving $f(x) = x^2 - 25$ in using The Newton-Raphson Method (with Python - SymPy):

In [1]:
```python
import sympy as sp

# Define the symbol x
X = sp.symbols('x')

# Define the equation and its derivative
equation = X**2 - 25
f = sp.lambdify(X, equation)
f_prime = sp.lambdify(X, sp.diff(equation, X))

def newton_raphson(x0, n, e):
    """
    Approximate the root of a function using the Newton-Raphson method.

    Parameters:
    - x0: Initial guess for the root
    - n: Maximum number of iterations
    - e: Tolerance for convergence

    Returns:
    - The approximated root if convergence is achieved within the specified tolerance,
      otherwise returns None.
    """
    for _ in range(n):
        x1 = x0 - f(x0) / f_prime(x0)
        if abs(f(x1)) < e:
            return x1
        x0 = x1
    return None

# Initial guess, maximum steps, and tolerance
initial_guess = 3.0
max_steps = 100
tolerance = 1e-6
```

```python
# Find the root using the Newton-Raphson method
root = newton_raphson(initial_guess, max_steps, tolerance)
print(f"Root found: {root}")
```

```
Root found: 5.000000002328306
```

RESULTS AND ANALYSIS

The calculations above can be tabulated as follows:

$METHOD$	$ROOT$	$ERROR(\%)$
$Completing\ Square$	5	0
$Newton - Raphson$	5.000000002328306	4.66×10^{-8}

Overall Observation:

- Both methods converge to a root close to 5.
- Completing Square Method gives an exact result, while Newton-Raphson Method provides a highly accurate approximation.
- Small error in the Newton-Raphson result suggests good convergence and reliability of the method for this equation.

CONCLUSION

In a nutshell, the Newton-Raphson method is applicable in finding the roots of the quadratic equation $f(x) = x^2 - 25$. The Completing Square method yielded an exact root of 5, while the Newton-Raphson method provided a highly accurate approximation with a root of 5.000000002328306.

The negligible error of 4.66×10^{-8} in the Newton-Raphson result signifies excellent convergence and reliability of the method for this particular equation. The comparison highlights the efficiency of the Newton-Raphson method in providing precise solutions for nonlinear equations, making it a valuable tool in numerical analysis.

REFERENCES

1. Anstee, M. Math 104 Newton's Method for Solving Equations. University of British Columbia. Retrieved from https://personal.math.ubc.ca/~anstee/math104/104newton-solution.pdf

2. BYJU'S Newton Raphson Method. Retrieved from https://byjus.com/maths/newton-raphson-method/

3. ScienceDirect Newton-Raphson Method. Retrieved from https://www.sciencedirect.com/topics/mathematics/newton-raphson-method