Yuchen Jin

CUS1188

Newton's Iteration

Introduction:

Roots of functions are input values that result in the function evaluating to zero. Graphically, roots can be located where the graph of a function intersects with the x-axis. How can calculators and computers precisely calculate roots of functions? Newton's Iteration is one possible way. Also known as Newton's Method and the Newton-Raphson Method, it is a numerical analysis method of closely approximating real roots of smooth or differentiable functions. It can be used to calculate roots of functions such as $f(x) = x^5 - x - 7$. This can be applied to calculate roots of numbers, such as $\sqrt{90}$ or $\sqrt[3]{75}$, and it can also be applied to precisely approximate division and reciprocals.

Concept:

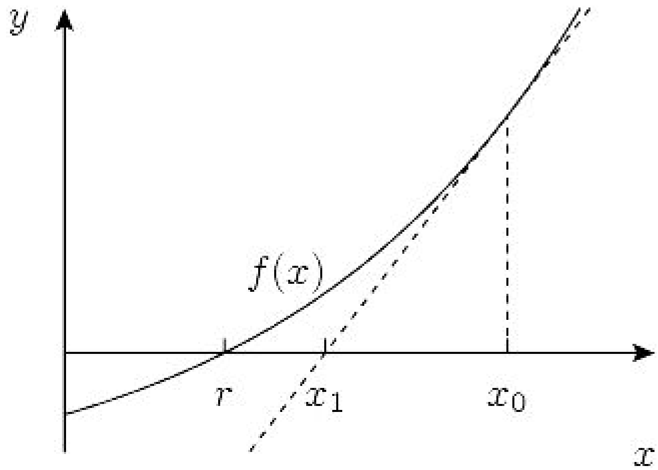
Newton's Iteration estimates roots of functions using its derivative. A derivative represents the slope of the tangent line at a particular point along the function. It begins with an initial given estimate of the root, x_0. Find the point at which the line tangent to the function at x_0 intersects with the x-axis. This becomes the new estimate, x_1 for the next iteration. This process is repeated until the estimate converges.

Recurrence Equation:

Below is the equation that describes the Newton Iteration algorithm. x_i represents the estimate for the root at each iteration.

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

This equation can be derived as follows:



(source)

The slope of the tangent line at $(x_0, f(x_0))$, $f'(x_0)$, can be expressed using the following equation: $f'(x_0) = \frac{f(x_0) - 0}{x_0 - x_1}$, obtained by inserting the points $(x_0, f(x_0))$ and $(x_1, 0)$ into the slope formula $m = \frac{y_2 - y_1}{x_2 - x_1}$.

Solve the slope equation for x_1:

$$(x_0 - x_1) f'(x_0) = f(x_0) - 0$$

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

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

If we derive the equation for x_2, we will end up with $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$ and so on for $x_3, x_4 \ldots$. The sequence can then be summarized using the Newton Iteration equation.

Example 1:

Given the following function: $f(x) = x^5 - x - 7$ and $x_0 = 2$, use Newton's Iteration to estimate the root of the function up to x_2.

$$f'(x) = 5x^4 - 1$$

$$x_1 = 2 - \frac{f(2)}{f'(2)} = 1.708860759$$

$$x_2 = 1.708860759 - \frac{f(1.708860759)}{f'(1.708860759)} = 1.56803704$$

$$x_3 = 1.56803704 - \frac{f(1.56803704)}{f'(1.56803704)} = 1.123$$

$$x_4 = - \frac{f()}{f'()}$$

Example 2:

Use Newton's Iteration to estimate $\sqrt[5]{1000}$. Start with $x_0 = 3.5$ and estimate the root up to x_4.

$$\sqrt[5]{1000} = 1000^{\frac{1}{5}} = x$$

$$x - 1000^{\frac{1}{5}} = 0$$

$$x^5 - 1000 = 0 = f(x)$$

$$f'(x) = 5x^4$$

$$x_1 = 3.5 - \frac{f(3.5)}{f'(3.5)} = 4.132778009$$

$$x_2 = 4.132778009 - \frac{f(4.132778009)}{f'(4.132778009)} = 3.991807286$$

$$x_3 = 3.991807286 - \frac{f(3.991807286)}{f'(3.991807286)} = 3.9811209306$$

$$x_4 = 3.9811209306 - \frac{f(3.9811209306)}{f'(3.9811209306)} = 3.981071707$$

Actual root (calculated on a calculator) = 3.981071706

Code Implementation using Divide and Conquer:

Time Complexity:

$O(f(n)\ logn)$

Issues with this algorithm:

The result is not guaranteed to converge. If the initial guess provided is too far from the root(s), the estimates may never converge. This can be handled by choosing a guess that is between the x values in which the function goes from negative to positive (or vice versa).

Works Cited

Salvy, Bruno. “CSE202 - Design and Analysis of Algorithms (2018-2019).” Course: CSE202 - Design and Analysis of Algorithms (2018-2019), https://moodle.polytechnique.fr/course/view.php?id=6782#section-4.

Weisstein, Eric W. "Newton's Method." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/NewtonsMethod.html

Walls, Patrick. “Newton's Method.” Mathematical Python, 10 Sept. 2019, https://www.math.ubc.ca/~pwalls/math-python/roots-optimization/newton/.