

Newton's Iteration

Presented:

Yuchen Jin

Background

The problem: How can we calculate the roots of a function?

Newton's Iteration (also known as Newton's Method or the Newton-Raphson Method) is a numerical analysis method of approximating the roots of a function.

Applications of this method:

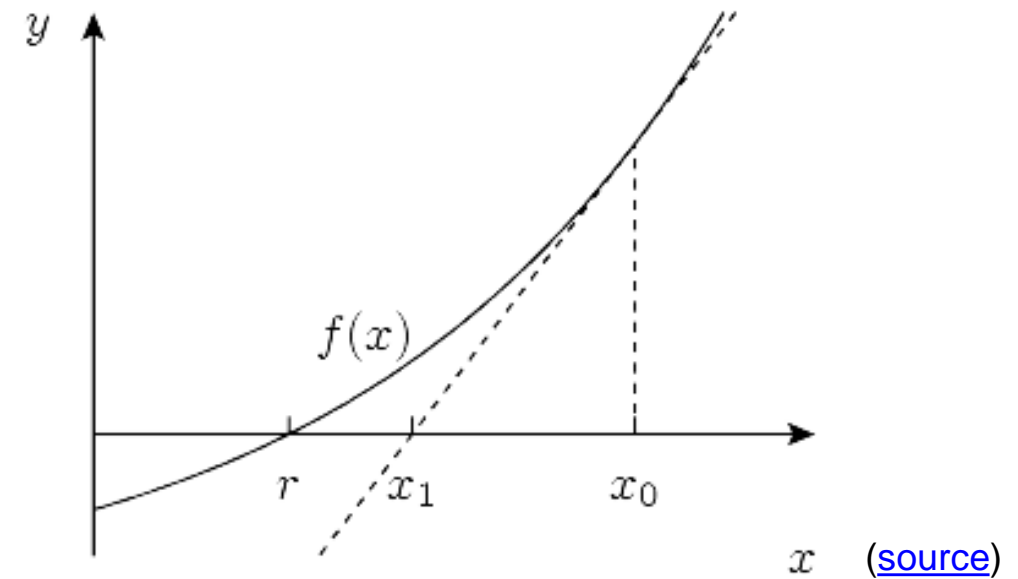
- ❑ Finding roots of functions
- ❑ Finding roots of numbers (ex: $\sqrt{77}$)
- ❑ Division
- ❑ Reciprocals (ex: the reciprocal of 77 is $\frac{1}{77}$)

Algorithm

1. Start with a given estimate, x_0 .
2. Find the tangent line of the function at x_0 .
3. The point at which this tangent line intersects with the x-axis becomes the next estimate, x_1 .
4. Repeat the process until the estimate starts to converge.

This process can be described with the following equation:

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



Deriving the equation

- The slope of the tangent line at $(x_0, f(x_0))$ can be expressed as the following equation: $f'(x_0) = (f(x_0) - 0) / (x_0 - x_1)$, obtained by inserting the points $(x_0, f(x_0))$ and $(x_1, 0)$ into the slope formula $m = (y_2 - y_1) / (x_2 - x_1)$.
- Solve the slope formula for x_1 :
- $(x_0 - x_1) f'(x_0) = f(x_0) - 0$
- $x_0 - x_1 = f(x_0) / f'(x_0)$
- $x_1 = x_0 - f(x_0) / f'(x_0)$
- If we derive the equation for x_2 , we will end up with $x_2 = x_1 - f(x_1) / f'(x_1)$ and so on for x_3, x_4, \dots . The sequence can be summarized with the Newton Iteration equation.

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

- $\sqrt[5]{1000} = 1000^{\frac{1}{5}} = x$
- $x - 1000^{\frac{1}{5}} = 0$
- $f(x) = x^5 - 1000$
- $f'(x) = 5x^4$
- $x_1 = 3.5 - f(3.5)/f'(3.5) = 4.132778009$
- $x_2 = 4.132778009 - f(4.132778009)/f'(4.132778009) = 3.991807286$
- $x_3 = 3.991807286 - f(3.991807286)/f'(3.991807286) = 3.9811209306$
- $x_4 = 3.9811209306 - f(3.9811209306)/f'(3.9811209306) = 3.981071707$
- Root calculated on a TI-84 calculator = 3.981071706

Division Using Newton's Iteration

Division (and the reciprocal of b) can be expressed as: $\frac{1}{b}$

We will express this as $\frac{1}{b} = x$.

$$b - \frac{1}{x} = 0$$

$$f(x) = b - x^{-1}$$

$$f'(x) = x^{-2}$$

Substitute $f(x)$ and $f'(x)$ into the recurrence equation and simplify:

$$x_{n+1} = x_n - \frac{b - x_n^{-1}}{x_n^{-2}}$$

$$x_{n+1} = x_n(2 - bx_n)$$

Example: Calculate $1/115$ using Newton's Iteration. Start with $x_0 = .001$

- $x_1 = .001(2 - 115 \cdot .001) = .001885$
- $x_2 = .001885(2 - 115 \cdot .001885) = .0033613791$
- $x_3 = .0033613791(2 - 115 \cdot .0033613791) = .0054233882$
- $x_4 = .0054233882(2 - 115 \cdot .0054233882) = .0074642653$
- $x_5 = .0074642653(2 - 115 \cdot .0074642653) = .0085212761$
- Quotient calculated on a T1-84 calculator = .0086956522

Quadratic Convergence

- ❑ Convergence is the rate at which the series described by the recurrence equation reaches its limit (and the limit in this case is our “answer”)
- ❑ Newton's Iteration has a quadratic rate of convergence. This means that for every iteration, its accuracy has essentially doubled.
- ❑ It can be expressed as:
 - ❑ $\frac{1}{a} - x_{n+1} = a\left(\frac{1}{a} - x_n\right)^2$

Code

```
from scipy.misc import derivative

# basic newton iteration algorithm
def newton(func, estimate, min_error=.0001):
    diff = func(estimate)/derivative(func, estimate)
    while abs(diff) > min_error:
        diff = func(estimate)/derivative(func, estimate)
        estimate = estimate - diff
        diff = func(estimate)/derivative(func, estimate)
    return estimate
```

Code

```
□ # newton iteration for division
□ def newton_div(a, min_error=.0001):
□     k = 0
□     while 10**k < a:
□         k += 1
□     init_estimate = 1/(10**k)
□     estimate = init_estimate * (2 - (a * init_estimate))
□     diff = estimate - init_estimate
□     while min_error < abs(diff):
□         old_estimate = estimate
□         estimate = estimate * (2 - (a * estimate))
□         diff = estimate - old_estimate
□     return estimate
```

Time Complexity

- Newton's Iteration: $O(\log n F(n))$
 - $\log n$ because of the quadratic convergence rate. Each iteration takes us halfway closer to the answer.
 - where $F(n)$ is the time boundary on the operation $f(x) / f'(x)$.

Thank you.