CS 3430: S19: SciComp with Py
Lecture 20

The Newton-Raphson Algorithm

Vladimir Kulyukin
Department of Computer Science
Utah State University

Review

Nth Taylor Polynomial

Given a function $f(x)$, the **nth Taylor polynomial of $f(x)$ at $x = 0$** is the polynomial $p_n(x)$ defined by

$$p_n(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + ... + \frac{f^{(n)}(0)}{n!}x^n.$$

This polynomial coincides with $f(x)$ up to the n-th derivative at $x = 0$ in the sense that $p_n(0) = f(0)$, $p_n'(0) = f'(0)$, ..., $p_n^{(n)}(0) = f^{(n)}(0)$.

Taylor Polynomial at x = a

Given a function $f(x)$, the **nth Taylor polynomial of** $f(x)$ **at** $x = a$ is the polynomial $p_n(x)$ defined by

$$p_n(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n.$$

This polynomial coincides with $f(x)$ up to the n-th derivative at $x = a$ in the sense that $p_n(a) = f(a)$, $p_n'(a) = f'(a)$, ..., $p_n^{(n)}(a) = f^{(n)}(a)$.

The Newton-Raphson Algorithm

Motivation

Many applications in scientific computing involve solving equations.

There is a function $f(x)$ and we must find a value of x, say $x = r$, such that $f(r) = 0$. This value of x is called a **root** of the equation $f(x) = 0$ (another frequently used term is a **zero** of the equation $f(x) = 0$).

When $f(x)$ is a polynomial, sometimes it is possible to factor and find zeros, sometimes it is impossible or not feasible computationally.

There are several methods for finding an approximation value of a zero to any desired degree of accuracy. The Newton-Raphson algorithm is one such method.

Outline of the Algorithm

Suppose that we know that a zero of $f(x)$ is approximately x_0.

The idea of the Newton-Raphson algorithm is to obtain an even better approximation of the zero by replacing $f(x)$ by its first Taylor at x_0. In other words,

$$p(x) = f(x_0) + \frac{f'(x_0)}{1}(x - x_0).$$

Since $p(x)$ closely approximates $f(x)$ near $x = x_0$, the zero of $f(x)$ should be close to the zero of $p(x)$.

Outline of the Algorithm

We can use the previous equation to solve $p(x) = 0$. Let's do it.

$$f(x_0) + f'(x_0)(x - x_0) = 0 \Rightarrow$$

$$f'(x_0)x = f'(x_0)x_0 - f(x_0) \Rightarrow$$

$$x = x_1 = x_0 - \tfrac{f(x_0)}{f'(x_0)}.$$

If x_0 is an approximation to the zero r, the number x_1 is a better approximation.

Outline of the Algorithm

We can obtain a better approximation x_2 to $x = r$ from x_1 in the same way as we obtained a new approximation x_1 from x_0. In other words,

$$x_2 = x_1 + \frac{f(x_1)}{f'(x_1)}.$$

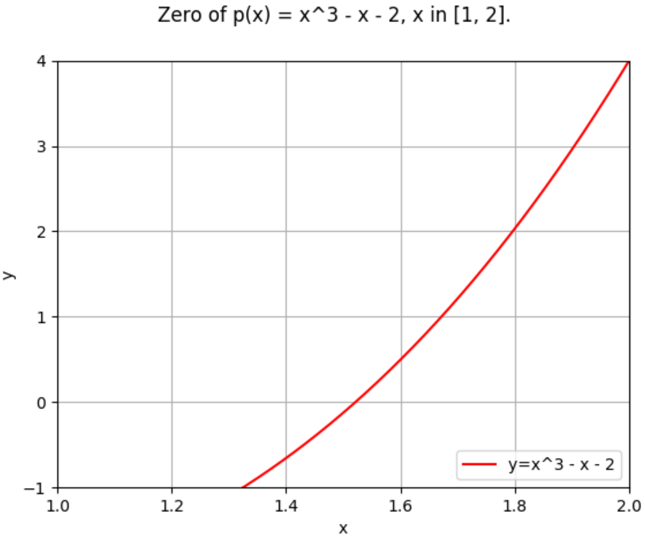
We may repeat this process over and over by obtaining the next approximation x_{n-1} from x_n:

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

Problem 1

Find a zero of the polynomial $f(x) = x^3 - x - 2$ by using the Newton-Raphson algorithm.

Solution



It looks like the zero of this polynomial is in $[1, 2]$.

Solution

Let's choose $x_0 = 1$. Then $f'(x) = 3x^2 - 1$.

$$x_1 = x_0 - \frac{x_0^3 - x_0 - 2}{3x_0^2 - 1} = 1 - \frac{1^3 - 1 - 2}{3(1)^2 - 1} = 1 - \frac{-2}{2} = 2.$$

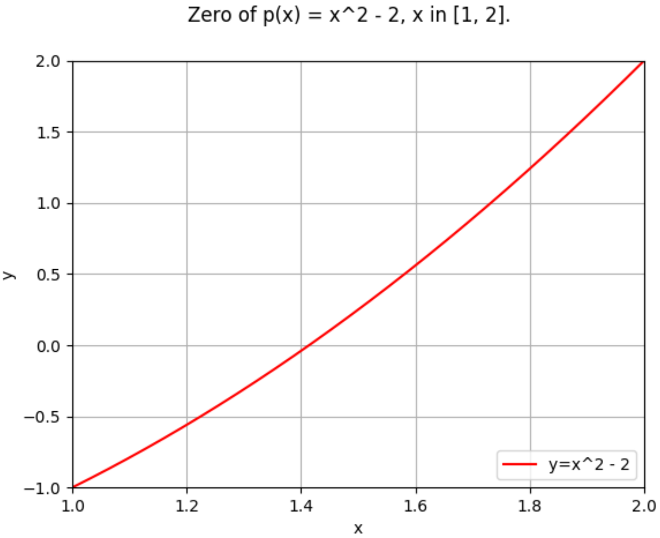
$$x_2 = 2 - \frac{2^3 - 2 - 2}{3(2)^2 - 1} = 2 - \frac{4}{11} = \frac{18}{11}.$$

$$x_3 = \frac{18}{11} - \frac{\left(\frac{18}{11}\right)^3 - \frac{18}{11} - 2}{3\left(\frac{18}{11}\right)^2 - 1} \approx 1.530.$$

Problem 2

Use four iterations of the Newton-Raphson algorithm to approximate $\sqrt{2}$.

Solution



It looks like the zero of $f(x) = x^2 - 2$ is in $[1, 2]$.

Solution

$\sqrt{2}$ is a zero of $f(x) = x^2 - 2$ and $f'(x) = 2x$. Thus, $\sqrt{2} \in [1, 2]$.
Let's take $x_0 = 1$.

$$x_1 = x_0 - \frac{x_0^2 - 2}{2x_0} = 1 - \frac{1^2 - 2}{2(1)} = 1 - \left(-\frac{1}{2}\right) = 1.5.$$

$$x_2 = 1.5 - \frac{(1.5)^2 - 2}{2(1.5)} \approx 1.4167.$$

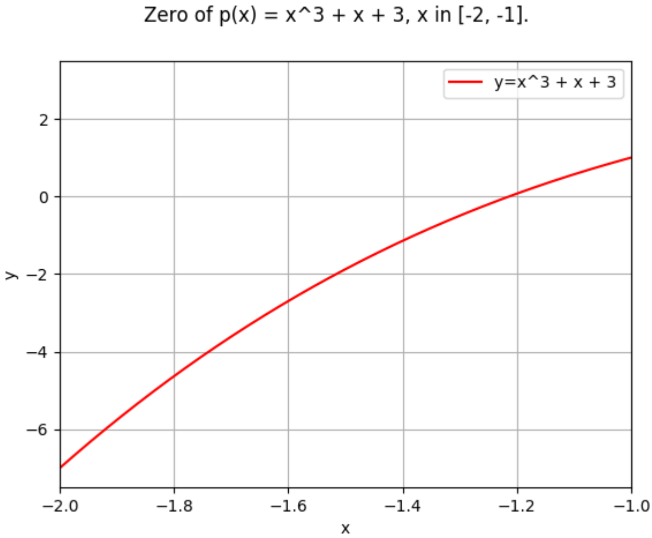
$$x_3 = 1.4167 - \frac{(1.4167)^2 - 2}{2(1.4167)} \approx 1.41422.$$

$$x_4 = 1.41422 - \frac{(1.41422)^2 - 2}{2(1.41422)} \approx 1.41421.$$

Problem 3

Approximate a zero of $f(x) = x^3 + x + 3$ with four iterations of the Newton-Raphson algorithm.

Solution



It looks like a zero of $f(x) = x^3 + x + 3$ is in $[-2, -1]$.

Solution

$f(x) = x^3 + x + 3$ and $f'(x) = 3x^2 + 1$. Let's take $x_0 = -1$.

$$x_1 = x_0 - \frac{x_0^3 + x_0 + 3}{3x_0^2 + 1} = -1 - \frac{(-1)^3 + (-1) + 3}{3(-1)^2 + 1} = -1.25.$$

$$x_2 = -1.25 - \frac{(-1.25)^3 + (-1.25) + 3}{3(-1.25)^2 + 1} \approx -1.2142857142857142.$$

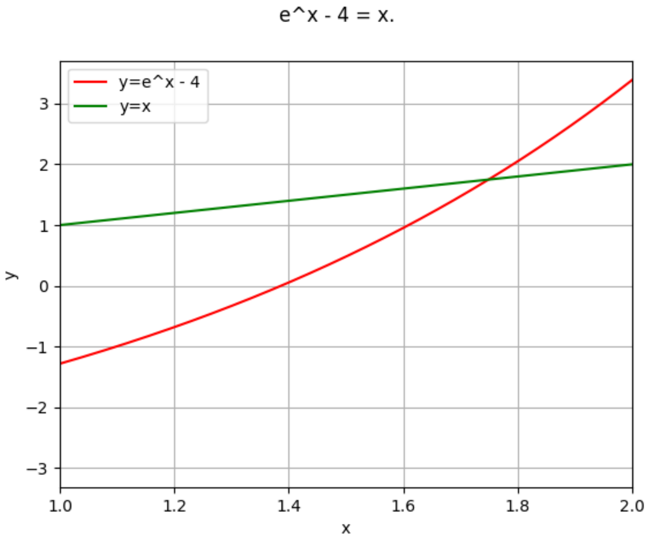
$$x_3 = -1.21429 - \frac{(-1.21429)^3 + (-1.21429) + 3}{3(1.21429)^2 + 1} \approx -1.213412180824357.$$

$$x_4 = -1.21341 - \frac{(-1.21341)^3 + (-1.21341) + 3}{3(-1.21341)^2 + 1} \approx -1.213411662764087\!4.$$

Problem 4

Approximate a positive solution to $e^x - 4 = x$.

Solution



It looks like the solution is in $[1, 2]$.

Solution

Let $f(x) = e^x - 4 - x$ and $f'(x) = e^x - 1$. Let's take $x_0 = 2$.

$$x_1 = x_0 - \frac{e^{x_0} - 4 - x_0}{e^{x_0} - 1} = 2 - \frac{e^2 - 4 - 2}{e^2 - 1} \approx 1.78.$$

$$x_2 = 1.78 - \frac{e^{1.78} - 4 - 1.78}{e^{1.78} - 4 - 1.78} \approx 1.75.$$

$$x_3 = 1.75 - \frac{e^{1.75} - 4 - 1.75}{e^{1.75} - 4 - 1.75} \approx 1.749.$$

References

1. L. Goldstein, D. Lay, D. Schneider, N. Asmar. *Calculus and its Applications*, Chapter 11. Pearson.
2. `www.python.org`.