4-2020

Equations for Which Newton's Method Never Works: Pedagogical Examples

Leobardo Valera

Martine Ceberio

Olga Kosheleva

Vladik Kreinovich

Comments:
Technical Report: UTEP-CS-20-31

Equations for Which Newton's Method Never Works: Pedagogical Examples

Leobardo Valera, Martine Ceberio, Olga Kosheleva, and Vladik Kreinovich

Abstract One of the most widely used methods for solving equations is the classical Newton's method. While this method often works – and is used in computers for computations ranging from square root to division – sometimes, this method does not work. Usual textbook examples describe situations when Newton's method works for some initial values but not for others. A natural question that students often ask is whether there exist functions for which Newton's method never works – unless, of course, the initial approximation is already the desired solution. In this paper, we provide simple examples of such functions.

1 Formulation of the Problem

Newton's method: a brief reminder. One of the most widely used methods for finding a solution to a non-linear equation $f(x) = 0$ is a method designed many centuries ago by Newton himself; see, e.g., [1]. This method is based on the fact that the derivative $f'(x)$ is defined as the limit of the ratio $\dfrac{f(x+h)-f(x)}{h}$ when h tends to 0. This means that for small h, the derivative is approximately equal to this ratio. In this approximation, $f'(x) \approx \dfrac{f(x+h)-f(x)}{h}$. Multiplying both sides by h, we get $f'(x)\cdot h \approx f(x+h)-f(x)$. Thus, adding $f(x)$ to both sides, we get

$$f(x+h) \approx f(x) + h \cdot f'(x). \tag{1}$$

Suppose that we know some approximation x_k to the desired value x. For this approximation, $f(x_k)$ is not exactly equal to 0. To make the value $f(x)$ closer to 0, it is therefore reasonable to make a small modification of the current approximation,

Leobardo Valera, Martine Ceberio, Olga Kosheleva, and Vladik Kreinovich
University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA
e-mail: leobardovalera@gmail.com, mceberio@utep.edu, olgak@utep.edu, vladik@utep.edu

i.e., take $x_{k+1} = x_k + h$. For this new value, according to the formula (1), we have $f(x_{k+1}) \approx f(x_k) + h \cdot f'(x_k)$. To get the value $f(x_{k+1})$ as close to 0 as possible, it is therefore desirable to take h for which

$$f(x_k) + h \cdot f'(x_k) = 0,$$

i.e., to take $h = -\dfrac{f(x_k)}{f'(x_k)}$. Thus, for the next approximation $x_{k+1} = x_k + h$, we get the following formula:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{2}$$

This is exactly what Newton has proposed.

If this method converges precisely – in the sense that we have $x_{k+1} = x_k$, then, from the formula (2), we conclude that $f(x_k) = 0$, i.e., that x_k is the desired solution. If this method converges approximately, i.e., if the difference $x_{k+1} - x_k$ is very small, then, from the formula (2), we conclude that the value $f(x_k)$ is also very small, and thus, we have a good approximation to the desired solution.

This method is still actively used to solve equations. In spite of this method being centuries old, it is still used to solve many practical problems. For example, this is how most computers compute the square root of a given number a, i.e., how computers compute the solution to the equation $f(x) = 0$ with $f(x) = x^2 - a$. For this function $f(x)$, we have $f'(x) = 2x$, thus Newton's formula (2) takes the form

$$x_{k+1} = x_k - \frac{x_k^2 - a}{2x_k}. \tag{3}$$

This formula can be further simplified if we take into account that $\dfrac{x_k^2}{2x_k} = \dfrac{x_k}{2}$ and thus, the formula (3) can be transformed into the following simplified form:

$$x_{k+1} = \frac{1}{2} \cdot \left(x_k + \frac{a}{x_k}\right). \tag{4}$$

The formula (4) is indeed faster to compute than the formula (3): both formulas require one division, but (3) also requires one multiplication (to compute x_k^2) and two subtractions, while (4) needs only one addition. (Both formula need multiplication or division by 2, but for binary numbers, this is trivial – just shifting by 1 bit to the left or to the right.)

The resulting iterative process (4) converges fast. For example, to compute the square root of $a = 2$, we can start with $x_0 = 1$ and get

$$x_1 = \frac{1}{2} \cdot \left(1 + \frac{2}{1}\right) = 1.5$$

and

$$x_2 = \frac{1}{2} \cdot \left(1.5 + \frac{2}{1.5}\right) = 1.4166\ldots,$$

i.e., in only two iterations, we already have the first three digits of the correct answer $\sqrt{2} = 1.414\ldots$

Newton's method also lies behind the way computers divide. To be more precise, computes compute the ratio $\frac{a}{b}$ by first computing the inverse $\frac{1}{b}$, and then multiplying a by this inverse. To compute the inverse, computers contain a table of pre-computed values of the inverse for several fixed values B_i, and, then, for $b \approx B_i$, use the recorded inverse $\frac{1}{B_i}$ as the first approximation x_0 in the Newton's method. In this case, the desired equation has the form $b \cdot x - 1 = 0$, i.e., here $f(x) = b \cdot x - 1$. The actual derivative $f'(x)$ is equal to b, i.e., ideally we should have

$$x_{k+1} = x_k - \frac{1}{b} \cdot (b \cdot x_j - 1). \tag{5}$$

This may sound reasonable, but since the whole purpose of this algorithm is to compute the inverse value $\frac{1}{b}$, we do not know it yet and thus, we cannot use the above formula directly. What we *do* know, at this stage, is the current approximation x_k to the desired inverse value $\frac{1}{b}$. So, a natural idea is to use x_k instead of the inverse value in the formula (5). Then, we get exactly the from of Newton's method that computers use to compute the inverse:

$$x_{k+1} = x_k - (b \cdot x_k - 1) \cdot x_k. \tag{6}$$

It should be mentioned that, similar to the expression (3), this expression can also be further simplified, e.g., to

$$x_{k+1} = x_k \cdot (2 - b \cdot x_k). \tag{7}$$

Both formulas (6) and (7) require two multiplications, but (7) is slightly faster to compute since this formula requires only one subtraction, while the formula (6) requires two subtractions.

Sometimes, Newton's method does not work. While Newton's method is efficient, there are examples when it does not work – such examples are usually given in textbooks, explaining the need for alternative techniques.

Sometimes, this happens because the values x_k diverge – i.e., become larger and larger with each iteration, never converging to anything. Sometime, this happens because the values x_k from a loop: we get $x_0, \ldots, x_{k-1}$, and then we again get $x_k = x_0$, $x_{k+1} = x_1$, etc. – and the process also never converges.

A natural question. The textbook examples usually show that whether Newton's method is successful depends on how close is the initial approximation x_0 to the actual solution:

- if x_0 is close to x, then usually, Newton's method converges, while
- if the initial approximation x_0 is far away from the actual solution x, Newton's method starts diverging.

A natural question – that students sometimes ask – is whether this is always the case, or whether there are examples when Newton's method never converges.

What we do in this paper. In this paper, we provide examples when Newton's method never converges, no matter what initial approximation x_0 we take – unless, of course, we happen to take exactly the desired solution x as the first approximation, i.e., unless $x_0 = x$.

2 First Example

Let us look for a simple example. Let us first look for examples in which the equation $f(x) = 0$ has only one solution. For simplicity, let us assume that the desired solution is $x = 0$.

Again, for simplicity, let us consider odd functions $f(x)$, i.e. functions for which $f(-x) = -f(x)$. Let us also consider the simplest possible case when the Newton's method does not converge: when the iterations x_k form a loop, and let us consider the simplest possible loop, when we have x_0, $x_1 \neq x_0$, and then again $x_2 = x_0$, etc.

How to come up with such a simple example. In general, the closer x_0 to the solution, the closer x_1 will be. If x_1 was on the same side of the solution as x_0, then:

- if $x_1 < x_0$, we would eventually have convergence, and
- if $x_1 > x_0$, we would have divergence,

but we want a loop. Thus, x_1 should be on the other side of x_0.

Since the function $f(x)$ is odd, the dependence of x_2 on x_1 is exactly the same as the dependence of x_1 on x_0. So:

- if $|x_1| < |x_0|$, we would have convergence, and
- if $|x_1| > |x_0|$, we would have divergence.

The only way to get a loop is thus to have $|x_1| = |x_0|$.

Since the values x_0 and x_1 are on the other solution of the solution $x = 0$, this means that we must have $x_1 = -x_0$. According to the formula (2), we have $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$. Thus, the desired equality $x_1 = -x_0$ means that $-x_0 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$. We want to have an example in which the Newton's process will loop for all possible initial values x_0 – except, of course, for the case $x_0 = 0$. Thus, the above equality must hold for all real numbers $x \neq 0$:

$$-x = x - \frac{f(x)}{f'(x)}. \tag{8}$$

Let us solve this equation. By moving the ratio the left-hand side and $-x$ to the right-hand side, we get

$$2x = \frac{f}{\dfrac{df}{dx}},$$

i.e., $2x = \dfrac{f \cdot dx}{df}$. We can now separate the variables x and f if we multiply both sides by df and divide both sides by f and by $2x$. As a result, we get $\dfrac{df}{f} = \dfrac{dx}{2x}$. Integrating both sides, we get $\ln(f) = \dfrac{1}{2} \cdot \ln(x) + C$, where C is the integration constant. Applying $\exp(z)$ to both sides of this equality, we get $f(x) = c \cdot \sqrt{x}$, where $c \stackrel{\text{def}}{=} \exp(C)$. Since we want an odd function, we thus get

$$f(x) = c \cdot \text{sign}(x) \cdot \sqrt{|x|}, \tag{9}$$

where $\text{sign}(x) = 1$ for $x > 0$ and $\text{sign}(x) = -1$ for $x < 0$.

Of course, if we shift the function by some value a, we get a similar behavior. Thus, in general, we have a 2-parametric family of functions for which the Newton's method always loops:

$$f(x) = c \cdot \text{sign}(x) \cdot \sqrt{|x-a|}. \tag{10}$$

Comment. Interestingly, the simplest example on which Newton's method never works – the example of a square root function $f(x)$ – is exactly inverse to the simplest example of a function $f(x) = x^2$ for which the Newton's method works perfectly.

3 Other Examples

Can we have other examples? Can we have similar always-looping examples for other functions, not just for the square root? Indeed, suppose that we have a non-negative function $f(x)$ defined for non-negative x, for which $f(0) = 0$ and for which, for each $x_0 > 0$, the next step of the Newton's method leads to the value $x_1 < 0$ – i.e., for which always

$$x - \frac{f(x)}{f'(x)} < 0. \tag{11}$$

This inequality can be reformulated as $f'/f < x$, i.e., as $\dfrac{\ln(f)}{\ln(x)} < 1$, i.e., the requirement that in the log-log scale, the slope is always smaller than 1.

We will also assume that the difference $x - \dfrac{f(x)}{f'(x)}$ monotonically depends on x.

How to design such lopping examples. We would like to extend the function $f(x)$ to negative values x in such a way that the Newton's process will always loop. For convenience, let us denote, for each $x > 0$, $F(x) \stackrel{\text{def}}{=} -f(-x)$, where $f(-x)$ is the desired extension. Then, for $x < 0$, we have $f(x) = -F(-x)$.

When we start with the initial value $x > 0$, the next iteration is $-y$, where we denoted

$$y = \frac{f(x)}{f'(x)} - x. \tag{12}$$

Then, if we want the simplest loop, on the next iteration, we should get back the value x, i.e., we should have

$$x = (-y) - \frac{f'(-y)}{f'(-y)}.$$

Substituting $f(x) = -F(-x)$ into this equality, we get

$$x = \frac{F(y)}{F'(y)} - y,$$

i.e., equivalently,

$$\frac{F'(y)}{F(y)} = \frac{1}{x+y}$$

and thus,

$$F'(y) = \frac{F(y)}{x+y}, \tag{13}$$

where $y(x)$ is determined by the formula (12).

We thus have a differential equation that enables us to reconstruct, step-by-step, the desired function $F(y)$ and thus, the desired extension of $f(x)$ to negative values.

Specific examples. For example, when $f(x) = x^a$ for some $a > 0$, the inequality (11) implies that $a < 1$. One can check that in this case, we can take $F(y) = y^{1-a}$, i.e., extend this function to negative values x as $f(x) = -|x|^{1-a}$.

In particular, for $a = 1/2$, we get the above square root example.

Acknowledgments

This work was supported in part by the US National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).

References

1. Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Massachusetts, 2009.