AEP 2 - Newton's Method

Alexander McColm

February 27, 2023

0 About this report

The goal is to document an exploration of Newton's Method in sufficient detail and clarity that a reader who is familiar with differential and integral calculus can easily follow along and understand.

It is also going to be submitted as an assignment for Calculus 2. Red letters and numerals like (Num.letter) indicate what task from the original assignment document is being done or was just done. These little markers are thus irrelevant to any reader besides my instructor. A quote from that original assignment document explains what I'm doing here:

> One of the most common tasks in mathematics is solving equations. You have learned a few techniques that allow you to solve relatively simple equations exactly using only "pencil and paper" (and careful thought). Most equations cannot be solved in this way.
>
> Mathematical software and advanced calculators have routines that can be used to solve these more complicated equations, but how do they work?
>
> Most of those methods are based on something called Newton's Method, and that is the topic of this AEP.

1 Towards Newton's Method

In this section I will show the process to construct a formula for the first improved estimate in terms of the initial guess. This formula can then be used to compute estimates of much greater accuracy.

1.1

The equation of the tangent line L to a differentiable function f at the point $(x_0, f(x_0))$ is as follows (1.a):

$$L(x) = f'(x_0)(x - x_0) + f(x_0)$$

1.2

I will solve for the x-intercept of the tangent line (1.b). Let this be represented by x_1.

$$y = f'(x_0)(x - x_0) + f(x_0)$$

Set $y = 0$, and substitute x_1 for x since we want a particular value of x.

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

Distribute the multiplication in the leftmost term on the right hand side.

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

Subtract $f'(x_0)(x_1)$ from both sides.

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

Multiply both sides by -1.

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

Divide both sides by $f'(x_0)$.

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

1.3

From this work, we can write a formula for x_{n+1} in terms of x_n (1.d):

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

2 Using Newton's Method to solve a test problem via spreadsheet

2.1 Test problem

We will solve for a solution to $x^2 = 9$, and we will try to solve for the solution $x = -3$. This makes a good test problem because given some math foundations it is easy to solve yourself.

2.2 Link

`https://nicollege-my.sharepoint.com/:x:/g/personal/amccolm_northislandcollege_ca/EVTagMVtpklPiJSrLUnFpPUBx9hJhW2rABTissPg553mpg?e=sfDnVQ` (2.a-i)

2.3 Derivative of f

The derivative of f: $f'(x) = 2x$ (2.c)

2.4 Table from Spreadsheet Used

This table contains the same results as the spreadsheet, truncated after they stop changing.

x	$f(x)$	$f'(x)$
-2.500000000000000	-2.750000000000000	-5.000000000000000
-3.050000000000000	0.302499999999998	-6.100000000000000
-3.000409836065570	0.002459184359045	-6.000819672131150
-3.000000027990440	0.000000167942657	-6.000000055980890
-3.000000000000000	0.000000000000000	-6.000000000000000
-3.000000000000000	0.000000000000000	-6.000000000000000

3 Using Newton's Method to solve $x^3 = 4x^2 + 1$

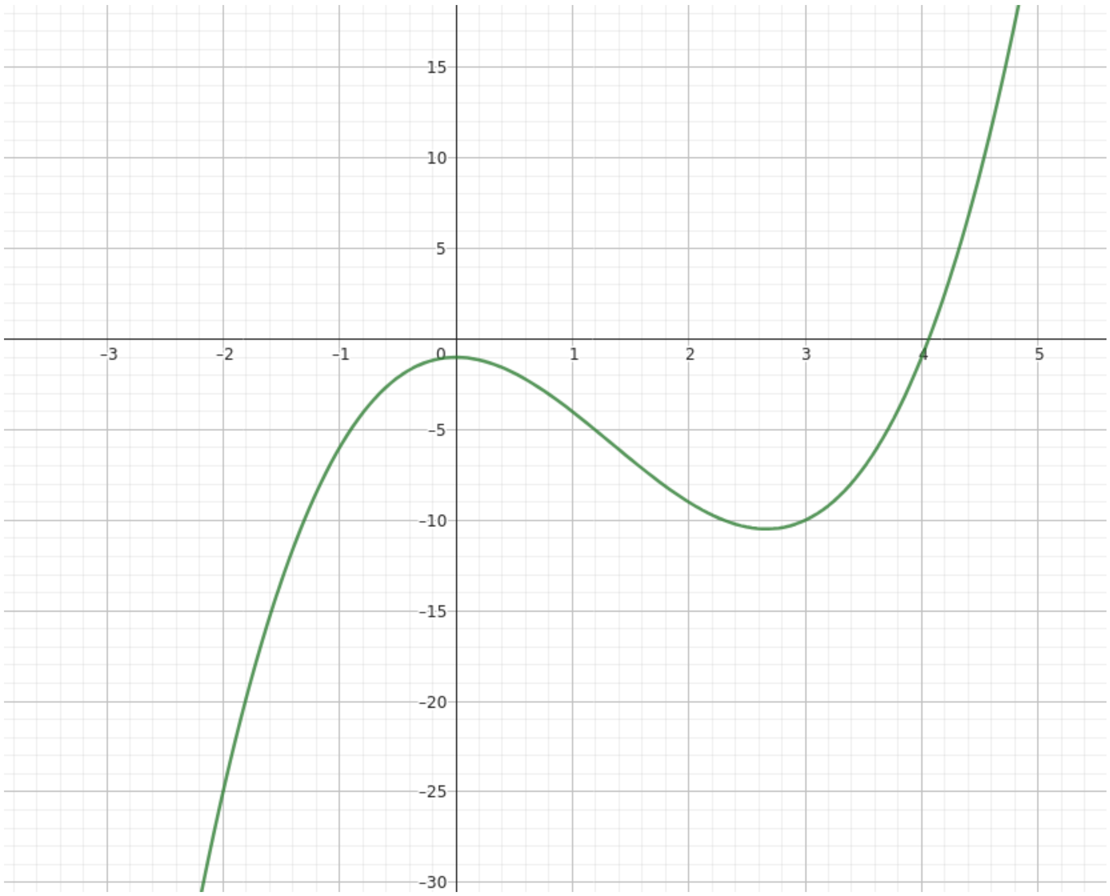
3.1 Opening move: finding a function f

The first thing we should do to solve $x^3 = 4x^2 + 1$ is define a function f such that $f = 0$ when x is one of the solutions. The function f whose x-intercepts coincide with solutions to the equation is given by $f(x) = x^3 - 4x^2 - 1$. Its derivative is $f'(x) = 3x^2 - 8x$.(3.a)

3.2 Geogebra file

`https://www.geogebra.org/calculator/zuawadsc` (3.b-c)

3.3 Graph of f



4 Using Newton's Method to estimate $\sqrt{17}$

4.1 Opening move

Again, we find a function f whose x-intercepts coincide with the solutions to the equation. In this case the function f equal to zero at the positive and negative numbers whose square is 17 is $f(x) = x^2 - 17$. (4)

4.2 Spreadsheet used

`https://nicollege-my.sharepoint.com/:x:/g/personal/amccolm_northislandcollege_ca/EeEjXopyGNFDquzUrP2XgngBZcweFrJZhK0vDonzLJaz4A?e=lTMQun`

4.3 Table

x	f(x)	f'(x)
1.00000000000000000000	-16.00000000000000000000	2.00000000000000000000
9.00000000000000000000	64.00000000000000000000	18.00000000000000000000
5.44444444444444000000	12.64197530864200000000	10.88888888888890000000
4.28344671201814000000	1.34791573469901000000	8.56689342403628000000
4.12610662758133000000	0.02475590217058250000	8.25221325516266000000
4.12310671696280000000	0.00000899946372356908	8.24621343392559000000
4.12310562561781000000	0.00000000000119371180	8.24621125123561000000
4.12310562561766000000	0.00000000000000000000	8.24621125123532000000

5 Consequences of a poor initial guess

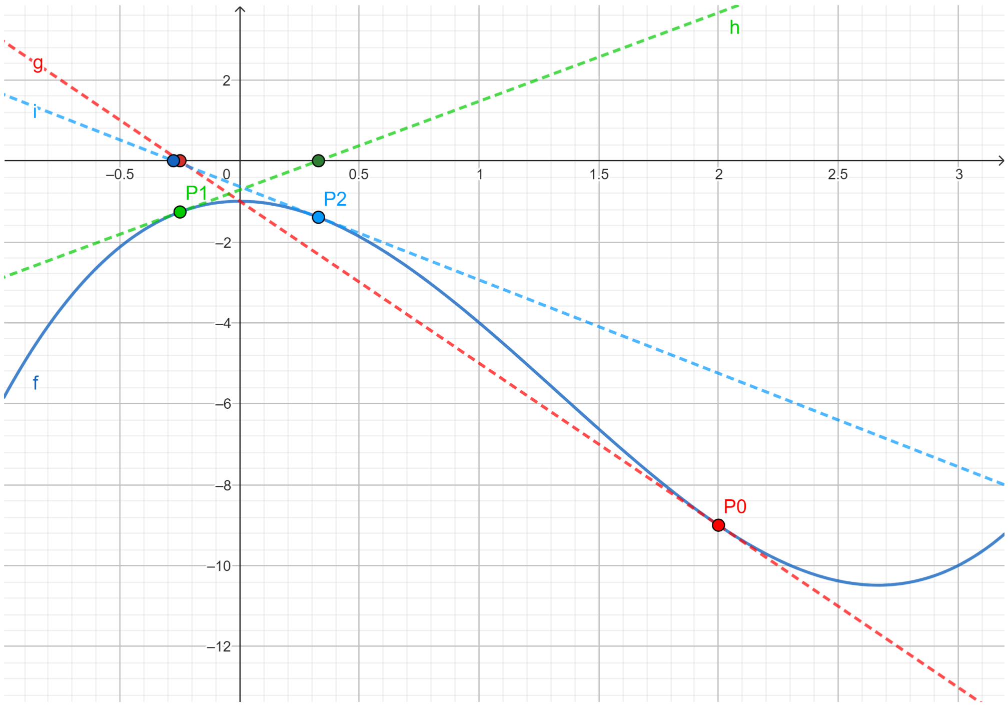
In order to see the consequences of a bad initial guess, I consider one of the equations from a previous section, $x^3 = 4x^2 + 1$. We will use two different initial guesses.

5.1 Initial guess of $x = 2$

With each step, the estimate is changing in undesired ways and is neither approaching a single value or approaching a correct one. (5.a)

5.2 Graphing the results of guessing $x = 2$

In the image, $P0$, $P1$, and $P2$ are points on the tangent lines obtained by the initial guess and the first and second estimates (respectively) given by Newton's Method.



(5.b)

5.3 Initial guess of $x = 0$

When I make this change in Excel, every cell from row 3 down throws a division by zero error. When I set x_0 to 0 in Geogebra, the corresponding tangent line becomes a horizontal line. A horizontal line, its derivative is of course zero. That first derivative being zero explains the division by zero errors. (5.c)

6 Examining the sentitivity of Newton's Method to the choice of x_0

6.1 Setting up

We will consider $f(x) = x^3 - x$. It is quick to determine that this has 3 x-intercepts at $x = -1, 0, 1$. Some observations about the process of solving by this method follow.

6.2 Convergence if $x_0 > \frac{1}{\sqrt{3}}$

It is not possible to find multiple roots from one initial guess with Newton's Method. As we iterate, we will approach a single root. It will be the closest root to our initial guess. When I use Newton's Method to solve for roots of the equation $x^3 - x = 0$, I have the following findings.

- $x_0 = -200$ or $x_0 = -2$ will both approach a solution of -1.
- $x_0 = 200$ or $x_0 = 2$ will both approach a solution of 1.
- $x_0 = -0.25$ or $x_0 = 0.25$ will both approach a solution of 0.

For a value greater than $\frac{1}{\sqrt{3}}$, the result of Newton's Method will get closer to 1 with each successive iteration, because this is the nearest place $f(x) = 0$ for any $x > \frac{1}{\sqrt{3}}$. (6.a)

6.3 Period-two points as x_1 do not obtain a solution

6.3.1 Proposition

If we use Newton's Method to solve for a solution to $f(x) = x^3 - x$ with an initial guess of $x_0 = \pm\frac{1}{\sqrt{5}}$, then A., $x_1 = \mp\frac{1}{\sqrt{5}}$, and B., $x_2 = \pm\frac{1}{\sqrt{5}}$. We proceed to direct proof.

6.3.2 Proof

Suppose that $f(x) = x^3 - x$, that its derivative is as given by $f'(x) = 3x^2 - 1$, and the formula for the next term in the series of estimates obtained by Newton's Method (x_{n+1} to follow x_n) is as given (6.b):

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

If we make an initial guess of $x_0 = \pm\frac{1}{\sqrt{5}}$, we can compute x_1 as follows.

$$x_1 = \pm\frac{1}{\sqrt{5}} - \frac{f(\pm\frac{1}{\sqrt{5}})}{f'(\pm\frac{1}{\sqrt{5}})}$$

Computing $f(\pm\frac{1}{\sqrt{5}})$.

$$f\left(\pm\frac{1}{\sqrt{5}}\right) = \left(\pm\frac{1}{\sqrt{5}}\right)^3 - \left(\pm\frac{1}{\sqrt{5}}\right) = \mp\frac{4\sqrt{5}}{25}$$

Computing $f'(\pm\frac{1}{\sqrt{5}})$.

$$f'\left(\pm\frac{1}{\sqrt{5}}\right) = 3\left(\pm\frac{1}{\sqrt{5}}\right)^2 - \left(\pm\frac{1}{\sqrt{5}}\right) = \pm(3/5 - 1) = \mp\frac{2}{5}$$

I divide the two fractions by flipping the divisor and multiplying.

$$\pm\frac{4\sqrt{5}}{25} \times \pm\frac{5}{2} = \pm\frac{2\sqrt{5}}{5}$$

Now we can finish computing x_1.

$$x_1 = \pm\frac{1}{\sqrt{5}} - \pm\frac{2\sqrt{5}}{5} = \mp\frac{1}{\sqrt{5}}$$

Therefore an initial guess of $x_0 = \pm\frac{1}{\sqrt{5}}$ causes x_1 to be equal to its opposite $\mp\frac{1}{\sqrt{5}}$. This proves part A of the proposition. Proving that $x_1 = \mp\frac{1}{\sqrt{5}}$ forces x_2 to be equal to $x_0 = \pm\frac{1}{\sqrt{5}}$ could be done by following the exact same process with opposite sign. This proves part B.

6.4

An estimate to the solution computed by Newton's Method with each of the following initial guesses (6.c):

- $x_0 = 0.577$
- $x_0 = 0.578$
- $x_0 = 0.460$
- $x_0 = 0.466$
- $x_0 = 0.44722$
- $x_0 = 0.44723$

are given in this spreadsheet where each initial guess has a page:
`https://nicollege-my.sharepoint.com/:x:/g/personal/amccolm_northislandcollege_ca/EUxSs6cidmpLpt0alyn_uo0B9ccdzxwQzy8sI9LazO5SoQ?e=Waxtbc`

The solutions are as follows:

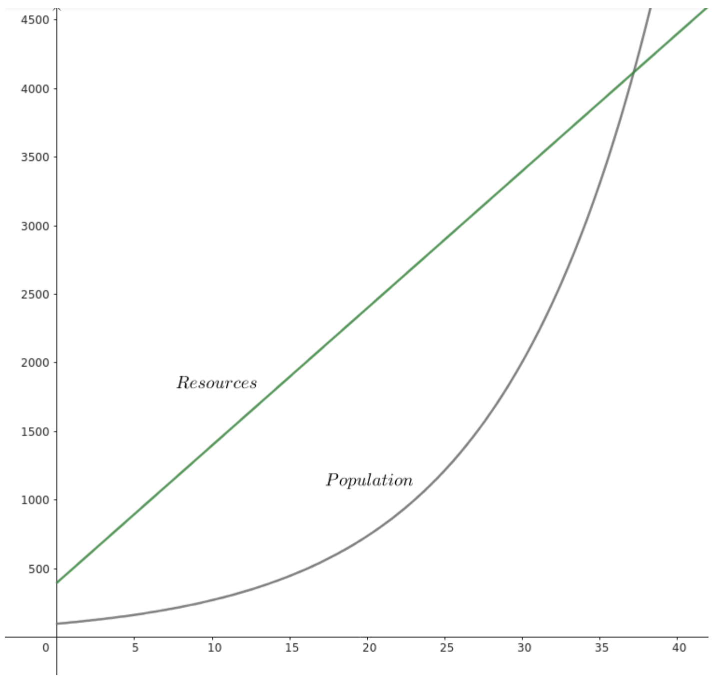
x_0	Solution found
0.577	-1
0.578	1
0.460	1
0.466	-1
0.44722	1
0.44723	-1

7 Solving a word problem by Newton's Method

7.1 Introduction

The problem is given as follows: "Thomas Malthus predicted doom for the human species when he argued that populations grow exponentially but their resources grow only linearly. Find the time when the population runs out of resources if the population grows according to $b = f(t) = 100e^{0.1t}$ and the resources grow according to $R = g(t) = 400 + 100t$. Malthus asserted that the population starves when $b = R$." I will estimate the solution highly accurately by Newton's Method. (7)

7.2 Graph of the problem



7.3 Creating a function whose x-intercept is the solution

We need to find the exact time when the population runs out of resources. I understand this as being equivalent to saying, "the time when the population is consuming exactly all of the resources available with no surplus and no want." The quantity of resources is greater than the population before that point, so I subtract the population from resources, and seek the time input for which that equation is equal to zero.

$$f(t) = 400 + 100t - 100e^{0.1t}$$

A quick differentiation gives:

$$f'(t) = 100 - 10e^{0.1t}$$

7.4 Forming an initial guess

From the graph in section 7.2, I see that the time we are looking for must occur for some t slightly larger than 35. So I will use an initial guess of $t_0 = 35$.

7.5 Spreadsheet used

`https://nicollege-my.sharepoint.com/:x:/g/personal/amccolm_northislandcollege_ca/EY2rXQtvYQpEjii5EXOMen8BnfEkkyUldEiZLUUlTjCWbw?e=XKgt3t`

7.6 Table

t	$f(t)$	$f'(t)$
35.000000000000000000	588.454804130769000000	-231.154519586923000000
37.545720521417200000	-117.021518389298000000	-327.159357053102000000
37.188030863865100000	-2.700287562637640000	-312.150337394915000000
37.179380264332900000	-0.001541675096632390	-311.793956810839000000
37.179375319801200000	-0.000000000502041075	-311.793753198062000000
37.179375319799600000	0.000000000000000000	-311.793753197996000000
37.179375319799600000	0.000000000000000000	-311.793753197996000000

7.7 Conclusion

The population will run out of resources in this model at approximately $t = 37.1793753197996$.

8 Appendix A: Program which estimates a square root by Newton's Method

8.1 About

On January 23rd while working on this AEP I saw an opportunity to squeeze some coding in and went for it. This program takes one natural number as a command line argument and prints an estimate of its square root to the shell. Exactly one month later, on February 23rd, I came back to this program and rewrote it in Python. The result is naturally much shorter and easier to read, and follows the C version.

8.2 C Code

An example command to use this program would be, if it is compiled to an executable named ”sqrt”:

```
> ./sqrt 16
4
```

```c
#define ACCURACY_ITERATIONS 100

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char * argv[])
{
   if (argc < 2)
   {
      fprintf(stderr, "Did not receive a number.\n");
      exit(EXIT_FAILURE);
   }
   // get a number from the user
   int s = atoi(argv[1]);

   // set an arbitrary starting guess for x
   double x = 1;

   // repeat until accuracy desired
   for (int i = 0; i < ACCURACY_ITERATIONS; i++)
   {
      // let x_{n+1} be the average of x_n and S/x_n
      x = (x + (s / x)) / 2;
   }
   // If the number is whole, we don't want to print it with leading 0s
   // So we use floor division to check
   double floor_x = floor(x);

   // If these values are the same, x is a whole number, eg. 16.000 == 16 evaluates true
   if (floor_x == x)
   {
      printf("%d\n",(int) x);
   }
   else // If not, there is a decimal we need to print.
   {
      printf("%f\n", x);
   }

   return 0;
}
```

8.3 Python 3 Code

An example command to use this program would be, if it is saved to a file named "sqrt.py":

```
> python3 sqrt.py 1001
31.63858403911275
```

```python
import sys, math

ITERATIONS = 100

if len(sys.argv) < 2:
    print('Did not receive a number!')
    sys.exit(1)

# Get a number from the user
s = int(sys.argv[1])

# Arbitrary starting guess
x = 1

# Repeat until accuracy desired
for i in range(ITERATIONS):
    x = (x + (s / x)) / 2

# Drop decimal, and check if this is equal to x anyways, printing result
floor_x = math.trunc(x)

if floor_x == x:
    print(floor_x)
else:
    print(x)
```

9 Appendix B: Newton's Method expressed as a step by step strategy

1. Find an $f(x)$ resembling that equation, such that $f(x) = 0$ where x is a solution to the equation.

2. Compute its derivative $f'(x)$.

3. Draw a rough sketch of the curve to inform an initial guess, often referred to as x_1.

4. In a table or spreadsheet, set up three columns. Allow extra width for decimals.

    - One for x values starting with x_1.
    - A second for $f(x)$ where x is the x-value in the column to the left.
    - A third for $f'(x)$ using the same x as the $f(x)$ column.

5. Enter your initial guess in the top left space, and the f and f' values obtained with that x in the spaces to the right of it.

6. For the next ($(n + 1)$th) row's x value, evaluate x_{n+1} using the formula given in section 1.3.

7. Compute the $f(x)$ and $f'(x)$ values for that row.

8. Repeat the previous two steps until the x value is of a desired accuracy. Note there are cases where the wrong guess could cause this method not to work, see section 6.