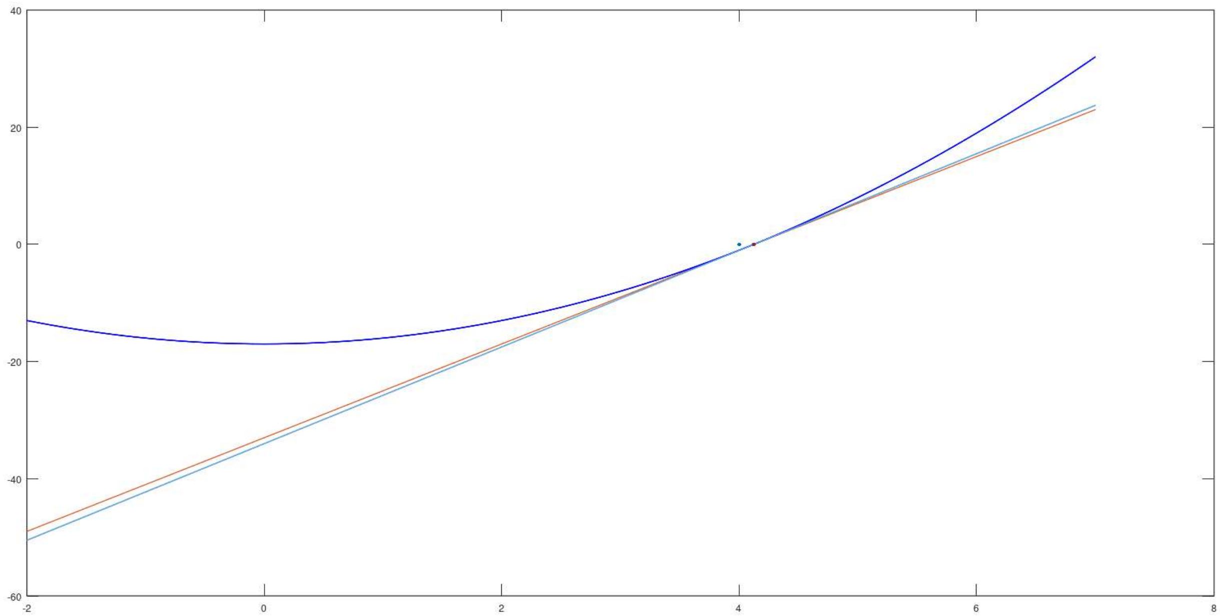
Math 2200 – Calculus I, Newton's Method

Throughout working on the Newton's Method project, I have had the chance to work on some important skills: first, this project helped me practice **Exploration**. I had never used Newton's method in the past, and had barely ever used mathematical computing tools, like MATLAB. This was a great project to practice Exploration skills, as it required me to not only do research about the method itself, but also downloading, reading and understanding a computer program focused on Mathematics, as well as exploring how to run it properly. This overlaps with another skill this project helped me with, **Research**. With exploration comes research, and in this case, I was able to research more about MATLAB and Octave, researching packages, mathematical notation, etc. It even made me lookup ways to create my own mathematical program! Lastly, this project emphasized **Problem Solving**, as mathematics on computers comes with hiccups, such as input problems: how to enter exponents, or the exponential function, how to enter square roots, etc., all of which were problems I had to solve.

Newton's Method

1) Solution of $x^2 - 17 = 0$

The approximate solution found using Newton's Method is $x \approx 4.1231$.



Name	Class	Dimension	Value
Nw	double	1x1	4.1231

The command used is: Nw=newto('x^2-17', 4, 10^(-3), -2, 7, 0.1);

Newton's Method only returned the positive solution. -4.1231 is also a solution of the equation.

Isaiah Graham & Xander Brunet

2) a)

Show that $x^2 - B = 0 \iff r_{n+1} = \frac{1}{2}\left(r_n + \frac{B}{r_n}\right)$

The recursive formula is : $r_{n+1} = r_n - \frac{f(r_n)}{f'(r_n)}$

$f(r_n) = r_n^2 - B$

$f'(r_n) = 2r_n$

We have : $r_{n+1} = r_n - \frac{r_n^2 - B}{2r_n} = r_n - \frac{r_n^2}{2r_n} + \frac{B}{2r_n} = r_n - \frac{r_n}{2} + \frac{B}{2r_n}$

$$= \frac{2r_n - r_n}{2} + \frac{B}{2r_n}$$

$$= \frac{r_n}{2} + \frac{B}{2r_n}$$

$$\boxed{= \frac{1}{2}\left(r_n + \frac{B}{r_n}\right)}$$

b)

Use $r_{n+1} = \frac{1}{2}\left(r_n + \frac{B}{r_n}\right)$ to find $\sqrt{23}$

Let $x^2 - 23 = 0$ with $B = 23$

Let $r_0 = 4$

$r_1 = \frac{1}{2}\left(4 + \frac{23}{4}\right) = \frac{39}{8} \approx 4.875$

$r_2 = \frac{1}{2}\left(\frac{39}{8} + \frac{23}{\frac{39}{8}}\right) = \frac{2993}{624} \approx 4.796$

$r_3 = \frac{1}{2}\left(\frac{2993}{624} + \frac{23}{\frac{2993}{624}}\right) = \frac{17913697}{3735264} \approx 4.796$

Since $r_2 = r_3$, we now know that $\boxed{\sqrt{23} \approx 4.796}$

c) Using different starting guesses affects the number of steps needed to come to a result: a worse initial guess means more iterations of the method. Newton's method also converges towards the root closest to the original guess, therefore if a function has multiple solutions, it will converge towards one of the two. In some cases, a guess can be so bad the function diverges.

Example with the equation $x^2 - 17 = 0$*:* trying with an initial guess of -4, we get $x \approx -4.1231$ which is the other root the first attempt in question 1) didn't give out.

The command used is: Nw=newto('x^2-17', -4, 10^(-3), -2, 7, 0.1);

Name	Class	Dimension	Value
Nw	double	1x1	-4.1231

We achieved the result in a total of 2 iterations:

```
>> Nw=newto('x^2-17', -4, 10^(-3), -2, 7, 0.1);
Symbolic pkg v3.2.1: Python communication link active, SymPy v1.10.1.
Guess = -4
f(guess) = -1
continue? y or n
Guess = -4.125
f(guess) = 0.015625
continue? y or ny
Done. Press enter.
```

However, trying with a value like 50, which is a worse guess:

```
>> Nw=newto('x^2-17', 50, 10^(-3), -2, 7, 0.1);
Guess = 50
f(guess) = 2483
continue? y or ny
Guess = 25.17
f(guess) = 616.5289
continue? y or ny
Guess = 12.9227
f(guess) = 149.9963
continue? y or ny
Guess = 7.1191
f(guess) = 33.6817
continue? y or ny
Guess = 4.7535
f(guess) = 5.596
continue? y or ny
Guess = 4.1649
f(guess) = 0.34647
continue? y or ny

Guess = 4.1233
```

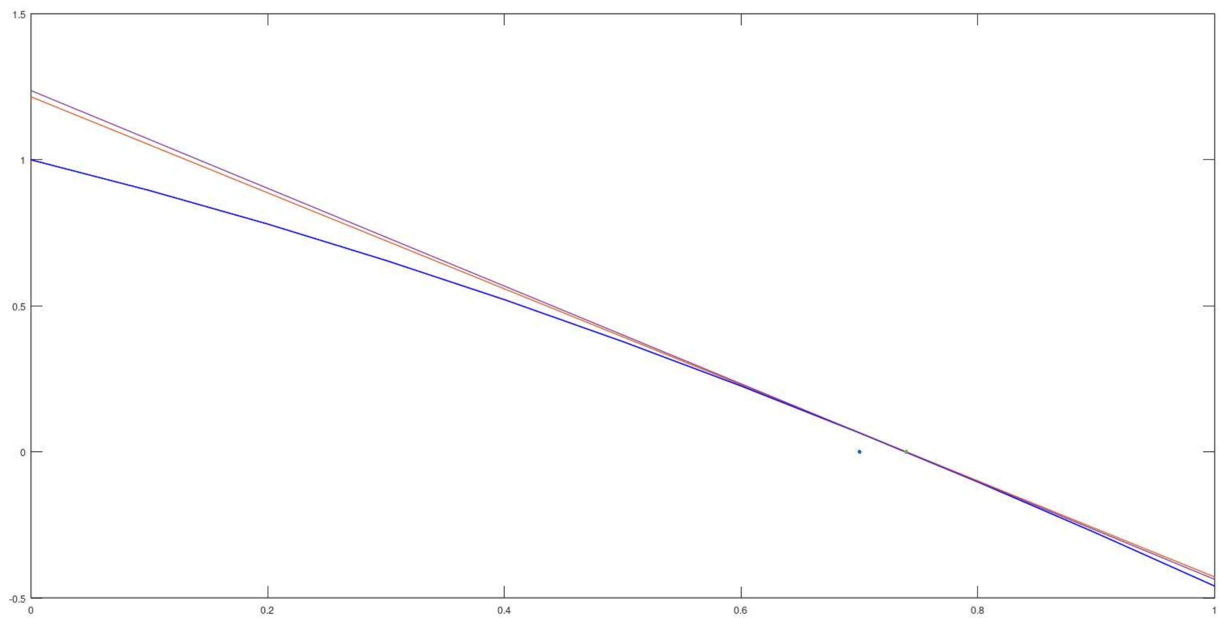
It takes a total of 6 iterations to find the result.

3) <u>Solution to $\cos(x) = x$</u>

Our initial guess on the interval [0,1] is $x \approx 0.7$.

Rewrite the equation as $\cos(x) - x = 0$.

The approximate solution found using Newton's Method is $x \approx 0.7394$.



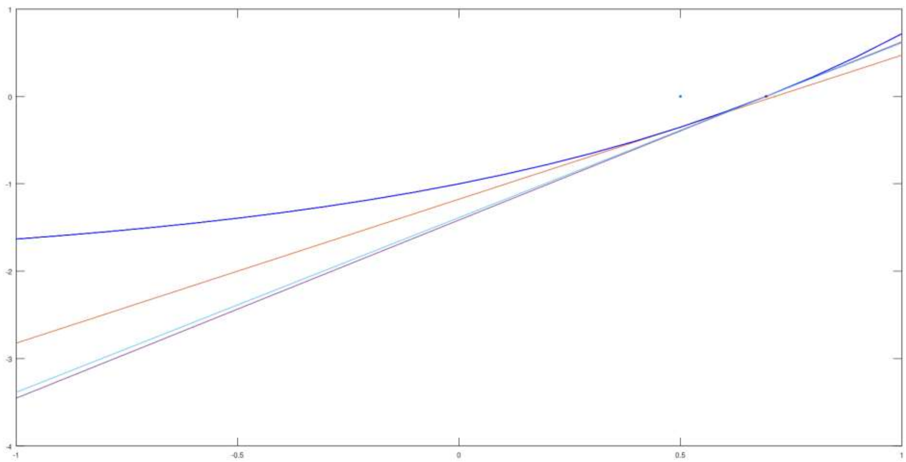
Name	Class	Dimension	Value
Nw	double	1x1	0.7394

The command used is: Nw=newto('cos(x)-x', 0.7, 10^(-3), 0, 1, 0.1);

4) <u>Approximating ln(2):</u>

An equation for which the solution is ln(2) is $e^x - 2 = 0$. A reasonable initial guess would be 0.5 because ln(1)=0, and the function ln increases less as x becomes bigger.

The command to enter is: Nw=newto('exp(x)-2', 0.5, 10^(-3), 1, 3, 0.1);



Name	Class	Dimension	Value
Nw	double	1x1	0.6933

With an initial guess of 0.5, it took 3 iterations to find the solution.

With an initial guess of -4, the function diverges away from the root, because the guess is so far from the actual solution.

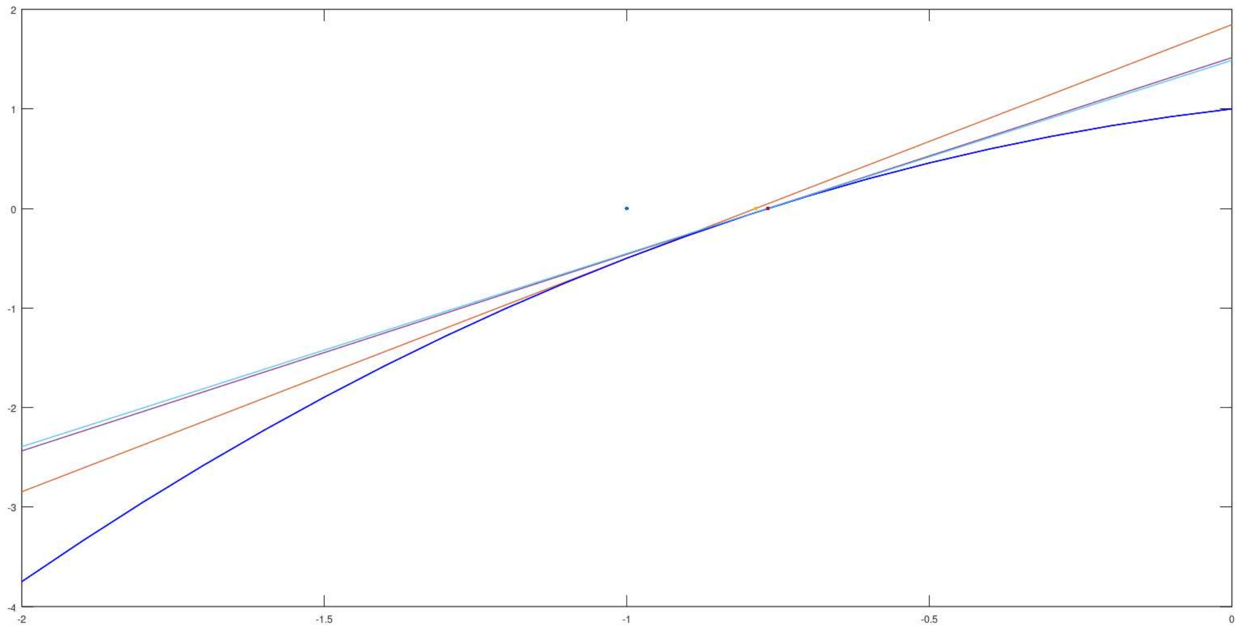
```
>> Nw=newto('exp(x)-2', (-4), 10^(-3), 1, 3, 0.1);
Symbolic pkg v3.2.1: Python communication link active, SymPy v1.10.1.
Guess = -4
f(guess) = -1.9817
continue? y or n
Guess = 104.1963
f(guess) = 1.785986424847984e+45
conwarning: opengl_renderer: data values greater than float capacity.  (1) Scale data, or (2) Use gnuplott
inwarning: called from
    newto at line 65 column 13
    newto at line 72 column 20u
e
? y or nwarning: opengl_renderer: data values greater than float capacity.  (1) Scale data, or (2) Use gnuplot
warning: called from
    newto at line 65 column 13
warning: opengl_renderer: data values greater than float capacity.  (1) Scale data, or (2) Use gnuplot
warning: called from
    newto at line 65 column 13
    newto at line 72 column 20
```

You can see the new approximate root being within the range of the 45th power of 10.

5) <u>Solutions of $x^2 = 2^x$.</u>

Let our initial guess be -1.

The command used is: Nw=newto('2^x-x^2', (-1), 10^(-3), 1, 3, 0.1);



Name	Class	Dimension	Value
Nw	double	1x1	-0.7668

The last root is approximately $x \approx -0.7668$.

Since the MATLAB script has not been modified, it has not been included in this document.

The meeting with the professor was done on Friday, July 5th at 3:00PM