Laboratory I.8  
Newton’s Method

Goals:
• The student will learn how to solve equations using Newton’s Method.
• The student will get an introduction to iteration and dynamical systems.

Before the Lab:

Newton’s Method is a technique which uses derivatives to solve equations that you can’t solve by hand (algebraically). An example is \( \cos x = x \). There are no algebraic tools which allow you to isolate \( x \) on only one side of that equation.

We’ll have you solve that particular equation in the lab; we’re going to work as our example a simpler equation that we can solve by hand so that we know what we’re doing. We’ll do \( x^2 = 2 \), and in particular we’ll pretend we don’t know the positive solution, \( x = \sqrt{2} \).

Solving \( x^2 = 2 \) is the same as solving \( x^2 - 2 = 0 \), so let’s graph \( y = x^2 - 2 \) and look where it crosses the \( x \)-axis:

![Graph of \( y = x^2 - 2 \)](image)

You can see the solution is just to the right of \( x = 1.4 \).

Newton’s Method works by taking an initial guess and improving it. Let’s use as our initial guess \( x_0 = 2 \). We’re going to add the tangent line to \( y = x^2 - 2 \) to the picture:

![Graph of tangent line at \( x = 1.4 \)](image)

The tangent line hits the \( x \)-axis at \( x = 1.5 \). Notice that this is much closer to the exact solution than our original guess, \( x_0 = 2 \). This, in essence, is Newton’s Method: the tangent line to the curve at a point will hit the \( x \)-axis closer to the exact root than the starting point.

Let’s check this with another picture. Our original guess of \( x_0 = 2 \) was “improved” to a guess of \( x_1 = 1.5 \). Let’s draw the tangent line to \( y = x^2 - 2 \) at \( x_1 = 1.5 \) and see what we get:
The new tangent line is so close to the function that we can hardly tell them apart. The intersection of the new tangent line with the x-axis is at \( x_2 = 1.4167 \); the exact value of the solution is \( x = 1.4142 \), so after just two iterations of Newton’s method, we have reduced our error to 0.0025. If we needed more decimal places of accuracy, we could continue, finding the tangent line for our current guess, \( x_2 = 1.4167 \).

So, this gives us **Newton’s Method (verbal/graphical version):** to find a root of \( f(x) = 0 \), start with an initial guess \( x_0 \). Find the tangent line to \( f(x) \) at \( x = x_0 \), and from that find where the tangent line hits the x-axis. That point of intersection is \( x_1 \). Continue following the same process from \( x_1 \) to get \( x_2 \), \( x_3 \) . . . until some estimate \( x_n \) is “close enough” to quit.

To use Derive (or any other computer tool) with Newton’s Method, we have to convert this verbal/graphical process to an algebraic one. So, let’s solve for \( x_1 \) as a function of \( x_0 \).

- Using the point-slope form, the equation for the tangent line to \( f'(x) \) at \( x = x_0 \) is
  \[
  y - f(x_0) = f'(x_0)(x - x_0)
  \]

- \( x_1 \) is the point at which this line hits the x-axis, so we plug in \( x_1 \) for \( x \) and 0 in for \( y \):
  \[
  0 - f(x_0) = f'(x_0)(x_1 - x_0)
  \]
  \[
  \frac{-f(x_0)}{f'(x_0)} = x_1 - x_0
  \]
  \[
  x_0 - \frac{f(x_0)}{f'(x_0)} = x_1
  \]

So this gives us **Newton’s Method (algebraic version):** to find a root of \( f(x) = 0 \), start with an initial guess \( x_0 \). Find a sequence of new guesses using the rule

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

to get new guesses from the previous guess. Continue until some estimate \( x_n \) is “close enough” to quit.

The definition of “close enough” varies according to different uses of Newton’s Method; we will use the following: **since we want** \( f(x) = 0 \), **stop when** \( |f(x_n)| < 0.01 \).
In The Lab

0. The most efficient way to get Derive to compute stuff for Newton’s Method is to

Declare Algebra State Simplification Branch Real
otherwise, Derive will evaluate the cube roots in #2b wrong

Declare Variable Domain x 0
(after the “OK”, choose “All”)
otherwise, Derive wants x0 to be x multiplied by 0

Author Newton(u, x, x0, n) := iterates(x-u/dif(u,x), x, x0, n)

What this does is produce the first n guesses after your initial guess x0. u stands for your
function, x is the name of your variable, x 0 is your initial guess, and n is the number of guesses
you want. For example, to reproduce the example in “Before the Lab”, you would first Author
the above, then

Author Newton(x^2 – 2, x, 2, 2)
Approximate the result

Now use Derive (or your calculator) to see if |1.4167^2 – 2| is indeed less than 0.01, or whether
you’d need to keep going.

1. Find all solutions for each of the following equations. Start by re-writing the equation so that
one side is 0, then graph the resulting function to see how many roots there are and a nearby
starting point for each root. Then use Newton’s method with your initial guesses to see what the
roots are. Again, stop when |f(xn)| < 0.01.

a) x^3 = 3

b) e^x = 3x

c) \cos x = x

2. Newton’s Method is pretty much the fastest way to solve equations numerically there is, and as
such it’s frequently used in industrial and applied situations when folks need to solve an equation.
However, Newton’s Method is not perfect: sometimes a bad initial guess can lead it astray,
sometimes it just doesn’t work at all.

a) Consider finding the roots of 4xe^{-x} = 1. What happens if your initial guess is x_0 = 1? Explain
what goes wrong either with a picture or algebraically. Pick a better initial guess for each root and
find them.

b) Consider finding the roots of \frac{3\sqrt{x}}{x} – 2 = 0. (OK, I know the root is x = 2, but let’s pretend we
don’t know for another example of what goes wrong.) Starting with x_0 = 1, what goes wrong
when you try to use Newton’s Method? Experiment to see if you can find any other value of x_0
that gets you close to x = 2. Print a graph of this function with the x-scale in the −10 to 10 range,
and sketch the tangent lines you get starting from an initial guess of x_0 = 1.
3. What happens if you try to use Newton’s Method when there’s no root at all? For example, try to find a root of \( x^2 + 1 = 0 \) with several initial guesses. Be sure to use \( n = 10 \) or so.

Ready for Lab?

None

After the Lab

Please turn in a disk with your work above. Also, fill in the chart below; turn in your graph from #2b; provide written responses to the questions in #2a and #2b; and explain from #3 what happens in general when you try to find a root for \( x^2 + 1 = 0 \).

<table>
<thead>
<tr>
<th>Solution for problem:</th>
<th>( n )</th>
<th>( x_0 )</th>
<th>( x_n )</th>
<th>( f \cdot x_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1(b)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1(c)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>