## 1170:Lab8

October 29th, 2013

## Goals For This Week

In this eighth lab we will explore Root Finding and Newton's Method

- Motivation
- Describing Newton's Method
- Code for Newton's Method


## 1 Motivation

Lets say that we have a function $\mathrm{f}(\mathrm{x})$ which models a system that we are interested in. Of particular interest would be when that function is equal to zero. For a population model this would be when the population would go extinct, for a chemical model it would represent when one of the substrates was used up. For a physical model this would represent when a physical quantity like force or velocity is equal to zero. Thus the problem which we would like to solve is finding the roots of a function. This is an exceptionally difficult problem for which an exact solution may not exist. As such we would like to create a method by which to approximate a function's root.

## 2 Describing Newton's Method

Last lab we learned how to approximate complex functions with polynomials. Polynomials are nice functions in that they are easy to differentiate, integrate, and for low degree polynomials it is easy to find the roots. Specifically it is very easy to find the root of a linear polynomial. Our goal then is to approximate a function near its root with a linear polynomial. Then find the root of the linear polynomial approximation and use that as our next guess for the root of the complicated function. Iterating in this fashion we will get closer and closer to the actual root of the function.

## 3 Code for Newton's Method

Suppose that we have a function $f(x)$, and that we can find its derivative $f^{\prime}(x)$. Let us then guess an initial value for the root, $\alpha$, such that $\alpha$ is close to the value of the root of $f(x)$. Then the tangent line:

$$
\begin{equation*}
y=f^{\prime}(\alpha)(x-\alpha)+f(\alpha) \tag{1}
\end{equation*}
$$

is a first order linear polynomial approximation of $f(x)$. Solving this for $y=0$ finds the root of our polynomial equation.

$$
\begin{equation*}
f^{\prime}(\alpha)(x-\alpha)+f(\alpha)=0 \tag{2}
\end{equation*}
$$

$$
\begin{gather*}
(x-\alpha)=\frac{-f(\alpha)}{f^{\prime}(\alpha)}  \tag{3}\\
x=\frac{-f(\alpha)}{f^{\prime}(\alpha)}+\alpha \tag{4}
\end{gather*}
$$

This gives us a new guess for the root, that is closer to it than our original guess. Repeating this process setting $\alpha=x$ will give us an even better estimate.

There are multiple features of this process that a computer can simplify. First it is an iterative process. We do the exact same thing over and over again. Next we only end up calling the same 2 functions, $f(x)$ and $\mathrm{f}^{\prime}(\mathrm{x})$. Here is what a code of Newton's Method looks like in R.

Let us consider a sample problem: finding the root of $f(x)=e^{x}-2$
Then $f^{\prime}(x)=e^{x}$
first let us plot $f(x)$ to make a guess for the root.

```
x<-seq(0,4,.01)
f<-function(x){exp(x)-2}
plot(x,f(x),type='l')
lines(x,0*x)
```

From the plot it looks like the graph crosses the x -axis near $\mathrm{x}=1$. Therefore let us use $\mathrm{x}=1$ as our initial guess

```
a<-1
fprime<-function(x){exp(x)}
a1<-(-f(a))/fprime(a)+a
a2<-(-f(a1))/fprime(a1)+a1
a3<-(-f(a2))/fprime(a2)+a2
a4<-(-f(a3))/fprime(a3)+a3
a5<-(-f(a4))/fprime(a4)+a4
error<-exp(a5)-2
```

Alternatively let us define a for loop
a<-1
n<-5
for (i in $1: n$ ) $\{a<-(-f(a)) / f$ prime (a) $+a\}$
Here n tells us how many times to go through the loop, and we rewrite a each time and use it as the new guess.

## 4 Assignment for this week

Consider the function $f(x)=x^{\frac{1}{3}}$ and an initial guess of $\mathrm{a}=1$.

1. Find what Newton's method gives us for the value of the root. Is this what you expect the root of $f(x)$ to be?

HINT: R can be very funny about roots of negative numbers, as such a for loop may not work. Consider instead solving Newton's method step by step like we did in the lab above and using the following syntax:
( $\left.a^{\wedge}(1 / 3)\right)$

Alternatively solve for Newton's method by hand to come up with a different recursion formula that doesn't involve cube roots of negative numbers.
2. What went wrong? Show in $R$ what Newton's method is doing by plotting the first couple of tangent lines: $y=f^{\prime}(a)(x-a)+f(a)$, and $y=f^{\prime}(a 1)(x-a 1)+f(a 1)$
3. What can you do to fix this problem?

