# 1170:Lab5

October 1st, 2013

**Goals For This Week**

In this fifth lab we will explore graphs of the Inverse function as well as Implicit Differentiation and Related Rates

- Plotting the inverse of a function
- Commands for creating a Movie in R
- Implicit Differentiation
- Visualizing Related Rates Problems with movies

## 1 Plotting The Inverse of a Function

Lets say I am given a function, like $f(x) = x^5 + x + 1$, this is a nice looking function, a quintic polynomial. However it turns out as a result of algrebraic group theory that it is impossible to analytically solve for the roots of this polynomial. This means that the best we can do is solve for the roots numerically (which we will be discussing in a couple of weeks). Another result that leads naturally from being unable to find an expression for the roots is that although we can prove that $f(x)$ has an inverse function, $f^{-1}(x)$, we can not solve for a formula for it.

To get a feel for what $f^{-1}(x)$ looks like we can use the fact that the graph of the inverse function is the graph of the original function $f(x)$ flipped over the line y=x.

To get started let us open up a .R file in gedit, and open R in the terminal. As we have been doing write you commands in the .R file in gedit and then copy and paste them into R.

To start with let us graph $f(x)$, $f^{-1}(x)$, and the line y=x for a Domain=[-5,5]

```
> x<-seq(-1,1,.01)
> y<-function(x){x^(5)+x+1}
> plot(x,y(x))
> lines(x,x)
> lines(y(x),x,type='l')
```

Hmm, this appears to have done what we want but isn't exactly too intuitive, we need to scale our x-axis and y-axis so that we have a square plot so we can see this better.

```
z<-range(c(x,y(x)))
plot(x,y(x),xlim=z,ylim=z)
lines(z,z)
lines(y(x),x)
```

Notice how we were able to flip our graph over the line y=x, all we had to do in R was flip around our x and y coordinate entries in our plot command. Pretty slick.

The line is $f^{-1}(x)$ and the points are $f(x)$, the diagonal line is y=x.

Now that we have a plot of $f^{-1}(x)$ we can sketch a graph of the derivative of $f^{-1}(x)$, even though we have no idea what that function is.

## 2 Commands for Creating a Movie in R

To visualize related rates problems we are going to learn how to create a ”movie” in R. To do this we will need to introduce a couple of new commands as well as a new concept, the for loop.

```
> Sys.sleep(timestep)
```

The Sys.sleep(timestep) command will cause R to pause in its calculations for however long you set the timestep to be. For example if timestep=1, R will pause for 1 second and then continue to the next command. We will use this command as a frame rate between images in our movie.

### 2.1 For Loops

For loops are a fundamental idea in coding. You use a for loop any time you want a computer to do a sequence of commands over and over again without you having to retype them each time. A for loop consists of the following:

- A counter variable
- Lower and upper bounds on the counter variable
- Commands that you want repeated that may or may not depend on the counter variable
- Closure of the Loop

For example consider the following code:

```
> for (i in 1:10) {x<-1+i}
```

Now let us dissect the code. first we have a for command followed by (i in 1:10). i is our counter variable, 1 is the lower bound, and 10 is the upper bound. x=1+i is the command that we will loop through each time

- The first time through the loop i=1 therefore x=1+1=2
- The second time through the loop i=1+1=2 therefore x=1+2=3
- ....
- The final time through the loop i=10 therefore x=1+10=11
- Once i=10 the loop breaks and we will step out of it.

### 2.2 The Movie Command

We can now combine the ideas of a for loop with a time delay to create a movie. Each time through the loop we need to create a new frame, or plot, and then have a time delay so that our eyes can process as the plots change.

For example:

```
> for (i in 1:10) {z<-seq(0,i,.01)
+ plot(z,z,xlim=range(0,11),ylim=range(0,11))
+ Sys.sleep(1)}
```

## 3 Related Rates

Suppose that we have a ladder that we are pushing up a wall. The ladder is 5 meters long, and the base of the ladder sits 4 meters away from the base of the wall. From Pythagorean theorem we know that the distance from the base of the ladder to the wall, x, and the top of the ladder from the ground, y, are related as

$$x^2 + y^2 = l^2 \tag{1}$$

Where l is the length of the ladder. Thus we have for our problem that:

$$x^2 + y^2 = 25 \tag{2}$$

Applying implicit differentiation with respect to t, time, we get the following equation

$$2x(t)\frac{dx}{dt} + 2y(t)\frac{dy}{dt} = 0 \tag{3}$$

As we are interested in how the ladder moves up the wall, let us solve this equation for $\frac{dy}{dt}$

$$\frac{dy}{dt} = -\frac{x}{y}\frac{dx}{dt} \tag{4}$$

Suppose that we push the ladder at a constant velocity, $\frac{dx}{dt} = -.1\frac{meter}{second}$ which is negative because we are moving towards the wall. Plugging this in for $\frac{dx}{dt}$ we get the following:

$$\frac{dy}{dt} = \frac{x}{10y} \tag{5}$$

Finally using Pythagorean theorem again we can solve for y in terms of x, $y = \sqrt{25 - x^2}$ and substitute this in to our equation:

$$\frac{dy}{dt} = \frac{x}{10\sqrt{25 - x^2}} \tag{6}$$

Now we have solved for $\frac{dy}{dt}$ in terms of x, which is constantly changing by $\frac{dx}{dt} = -0.1\frac{m}{s}$ and which starts at an initial value of $x_0 = 4m$

## 4 Visualizing Related Rates with Movies

It turns out that we did not need to solve the related rates problem to be able to visualize a plot of the solution. By Pythagorean theorem if I know my x-value I can solve for my y-value, the relation between x and y is a line, since ladders are straight, and i know how x moves with time, so i can solve for x at any time: $x(t) = x_0 + \frac{dx}{dt}t = 4 - 0.1t$.

The picture of the ladder that I want to plot is a line, with $y - intercept = y(t)$, and $slope = -\frac{y(t)}{x(t)}$

$$y = -\frac{y(t)}{x(t)}x + yt \tag{7}$$

and since y(t) and x(t) are related through Pythagorean theorem $y(t) = \sqrt{25 - x(t)^2}$ we get the following equation for the ladder:

$$y = -\frac{\sqrt{25 - x(t)^2}}{x(t)}x + \sqrt{25 - x(t)^2} \tag{8}$$

The next thing we need to do is figure out over what domain we should be plotting our ladder. We want it to start at the wall, so our left bound should be x=0, and we want it to end when it is touching the floor, so our right bound should be x(t). These will be our values for x.

We now have everything that we need to create the movie. Let us run it for 30 seconds. We will call x(t)=xt, x=x, y=y, and i will be our counter variable and it is related to time in the following way: time=i-1, so the first step will be when time=0, and the last step will be when time=30

```
> for (i in 1:31) {xt<- 4-(0.1)*(i-1)
+ x<- seq(0,xt,.01)
+ y<-function(x){-(sqrt(25-xt^2)/xt)*x+sqrt(25-xt^2)}
+ plot(x,y(x),xlim=range(0,5),ylim=range(0,5),type='l')
+ Sys.sleep(1)}
```

## 5 Assignment for this week

Suppose that we have a 20 foot ladder propped up against a wall such that its initial height against the wall is 18 feet. Do to a nonlinear frictional resistance the ladder slides down the wall at a constant velocity of 0.5 feet per second.

1. Create a movie showing the ladder's progression for 30 seconds (i=1 to 31 like in the example), submit your code in a .R file
2. How far is the base of the ladder from the wall after 30 seconds?
3. What is the velocity of the base of the ladder after 30 seconds? i.e. what is $\frac{dx}{dt}$ when t=30?