Newton's Method

Morgan de Ferrante

January 8, 2018

Newton's Method Formula

Newton's method is a root-finding method that uses the derivative to solve for a given function.

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

The method begins with an initial value x_0, which needs to be sufficiently close to the true value of the root. Thus iteration of the method begins with the following:

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

The following newtons_method function generates a list of the iterative values, and a plot of x_n versus $f(x_n)$. The newtons_method function takes in a function object type (f_and_fprime) with both the given function for which we wish to estimate roots and its derivative function, then the initial x_0, and the number of desired iterations.

```
library(ggplot2)

newtons_method <- function(f_and_fprime, x0, iter = 5) {
  # applies Newton-Raphson to find x such that f_and_fprime(x)[1] == 0
  # f_and_fprime is a function of x. it returns two values, f(x) and f'(x)
  # x0 is the starting point

  # df_points is used to track each update
df_points <- data.frame(x1 = numeric(0),
                        y1 = numeric(0),
                        x2 = numeric(0),
                        y2 = numeric(0))
  xnew <- x0
  cat("Starting value is:", xnew, "\n")

  # the algorithm which generates each xn value
  for (i in 1:iter) {
    xold <- xnew
    f_xold <- f_and_fprime(xold)
    xnew <- xold - f_xold[1]/f_xold[2]
    #f_xold[1] is the value of the original function at xold
    #f_xold[2] is the value of its derivative at xold
    cat("Next x value:", xnew, "\n")

    df_points[2*i - 1,] <- c(xold, 0 , xold, f_xold[1]) # vertical segment
    df_points[2*i,] <- c(xnew , 0 ,xold ,f_xold[1])   # tangent segment
  }

  # determines the limits for the graph
```

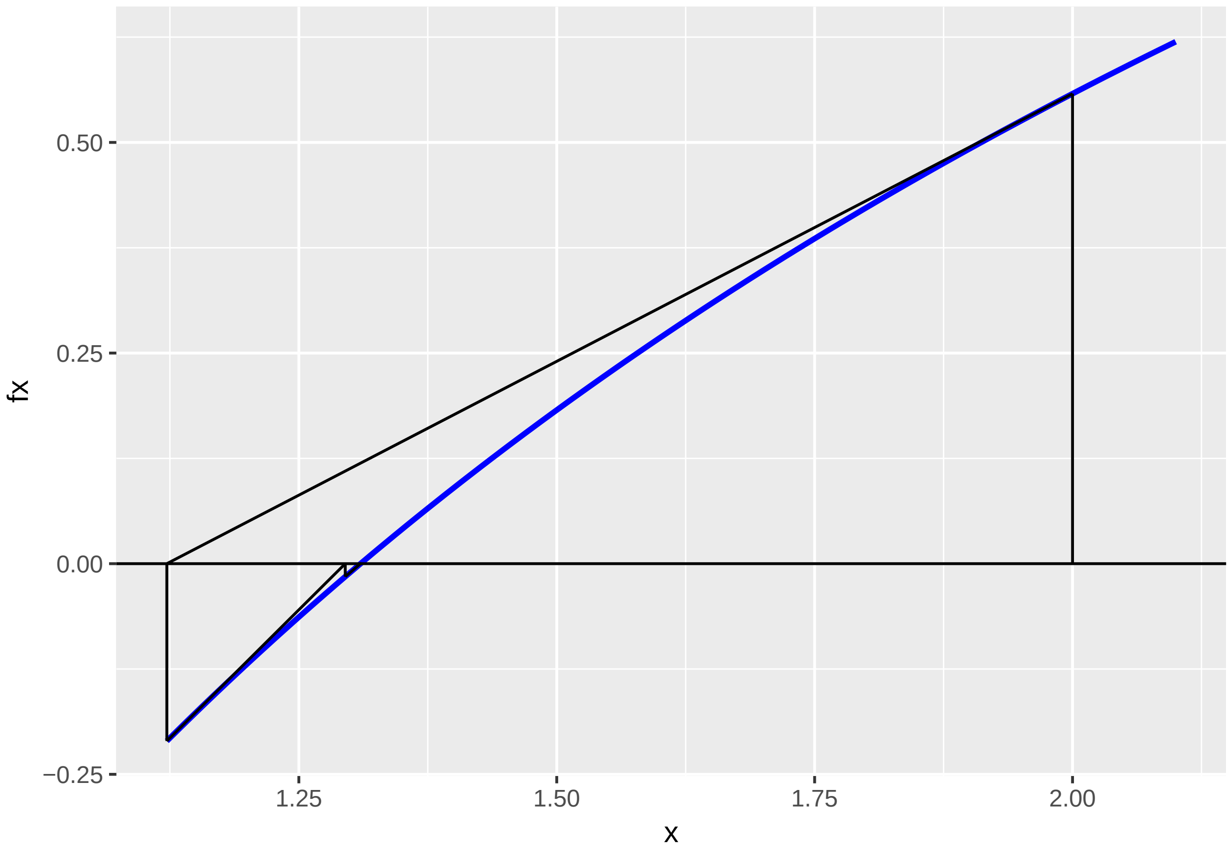
```r
  x_start <- min(df_points$x1, df_points$x2, x0 - .1) # start is the min of these values
  x_end <- max(df_points$x1, df_points$x2, x0 + .1)  # end is the max of these values
  x <- seq(x_start, x_end, length.out = 200)
  fx <- rep(NA, length(x))
  for (i in seq_along(x)) {
    fx[i] <- f_and_fprime(x[i])
  }
  function_data <- data.frame(x, fx)

  # uses ggplot to plot the function and the segments for each iteration
  p <- ggplot(function_data, aes(x = x, y = fx)) +
    geom_line(colour = "blue", size = 1) +
    geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2), data = df_points) +
    geom_abline(intercept = 0, slope = 1) +
    geom_abline(intercept = 0, slope = 0)

  print(p)
  return(cat("The estimated root for the function is", xnew))
}
```

Example

This example estimates the zero for the function $f(x) = log(x) - e^{-x}$ using the initial value of $x_0 = 2$.

```r
f_and_fprime <- function(x){
  f <- log(x) - exp(-x)   # f(x)
  fprime <- 1/x + exp(-x)  # f'(x)
  return(c(f, fprime))
}
newtons_method(f_and_fprime, 2, iter = 8)
```

```
## Starting value is: 2
## Next x value: 1.12202
## Next x value: 1.294997
## Next x value: 1.309709
## Next x value: 1.3098
## Next x value: 1.3098
## Next x value: 1.3098
## Next x value: 1.3098
## Next x value: 1.3098
```




```
## The estimated root for the function is 1.3098
```