PM520–HOMEWORK 2

Chengliang Dong Thursday 12^{th} February, 2015

1 Introduction

In this homework, we are looking at two different methods for finding the root. Suppose that $f : R \to R$ is a continuous function. A root of f is a solution to the equation $f(x) = 0$. That is, a root is a number $a \in R$ such that $f(a) = 0$. If we draw the graph of our function, say $y = f(x)$, which is a curve in the plane, a solution of $f(x) = 0$ is the x-coordinate of a point at which the curve crosses the x-axis.
The first method is Newton-Raphson method. Suppose our function f is differentiable with continuous derivative f' and a root a. Let $x_0 \in R$ and think of x_0 as our current 'guess' at a. Now the straight line through the point $(x_0, f(x_0))$ with slope $f'(x_0)$ is the best straight line approximation to the function $f(x)$ at the point x_0 (this is the meaning of the derivative). The equation of this straight line is given by $f'(x_0) = \frac{f(x_0)-y}{x_0-x}$. Now this straight line crosses the x-axis at a point x_1, which should be a better approximation than x_0 to a. To find x_1 we observe $f'(x_0) = \frac{f(x_0)-0}{x_0-x_1}$ and so $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$ In other words, the next best guess x_1 is obtained from the current guess $x0$ by subtracting a correction term $\frac{f(x_0)}{f'(x_0)}$. Now that we have x_1, we use the same procedure to get the next guess $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$. Or in general:

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

The second method is the secant method. A problem with the Newton-Raphson algorithm is that it needs the derivative f'. If the derivative is hard to compute or does not exist, then we can use the secant method, which only requires that the function f is continuous. Like the Newton-Raphson method, the secant method is based on a linear approximation to the function f. Suppose that f has a root at a. For this method we assume that we have two current 'guesses', x_0 and x_1, for the value of a. We will think of x_0 as an older guess and we want to replace the pair x_0, x_1 by the pair x_1, x_2, where x_2 is a new guess. To find a good new guess x_2 we first draw the straight line from $(x_0, f(x_0))$ to $(x_1, f(x_1))$, which is called a secant of the curve $y = f(x)$. Like the tangent, the secant is a linear approximation of the behaviour of $y = f(x)$, in the region of the points x0 and x1. As the new guess we will use the x-coordinate x_2 of the point at which the secant crosses the x-axis. Now the equation of the secant is given by $\frac{y-f(x_1)}{x-x_1} = \frac{f(x_0)-f(x1)}{x_0-x_1}$ and so x_2 can be found from $\frac{0-f(x_1)}{x_2-x_1} - \frac{f(x_0)-f(x_1)}{x_0-x_1}$, which gives, $x_2 = x_1 - f(x_1)\frac{x_0-x_1}{f(x_0)-f(x_1)}$. Repeating this we get a second-order recurrence relation (each new value depends on the previous two):

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \tag{2}$$

In the current homework, we are interested in comparing these two methods in their ability to find roots for $cos(x) - x$ and $log(x) - exp(-x)$. We are also interested in how different initialization value impacts the number of iterations, speed and final estimated roots of these two different methods.

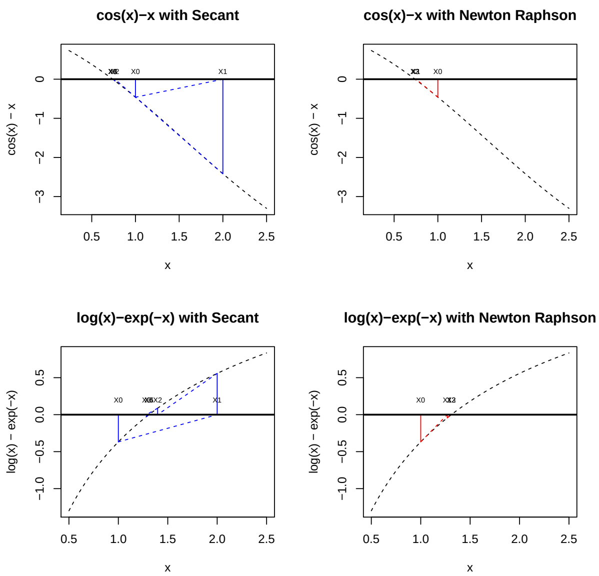
2 Method

Two functions were used to investigate the difference between Newton-Raphson and secand methods on thier ability to find the roots for them. The first function was $cos(x) - x$ and the second one was $log(x) - exp(-x)$. First, to play with the toy initialization values given by the professor, we implemented these two methods with $x_0 = 1, x_1 = 2$. We drew convergence plot and demonstrated the difference in pattern of convergence of these two methods. Next, to investigate the effect of initialization value on number of iterations, speed and final estimated roots of these two different methods, we generated 200 different initialization values based on the true solution of these two functions. For $cos(x) - x$, we used 0.739 as its true solution and generated 200 different initialization values as follows, $0.739 \times 1.1^0, 0.739 \times 1.1^1, 0.739 \times 1.1^2, 0.739 \times 1.1^3 \ldots 0.739 \times 1.1^{199}$. For $log(x) - exp(-x)$, we used 1.310 as its true solution and generated 200 different initialization values as follows, $1.310 \times 1.1^0, 1.310 \times 1.1^1, 1.310 \times 1.1^2, 1.310 \times 1.1^3 \ldots 1.310 \times 1.1^{199}$. Then for each initialization value, we implemented these two methods and investigated how number of iterations, speed and final estimated roots of these two different methods vary as we set initialization value further away from the true solution.

3 Result

3.1 Toy example

Figure 1 demonstrated the simple comparison between these two methods using the toy initialization values given by the professor. From this figure we can see the convergence pattern of these two patterns are similar at initialization values of $x_0 = 1, x_1 = 2$.



(a) Simple comparison

Figure 1: Simple comparison between these two methods $x_0 = 1, x_1 = 2$

3.2 In depth analysis

3.2.1 number of iterations and final estimated roots

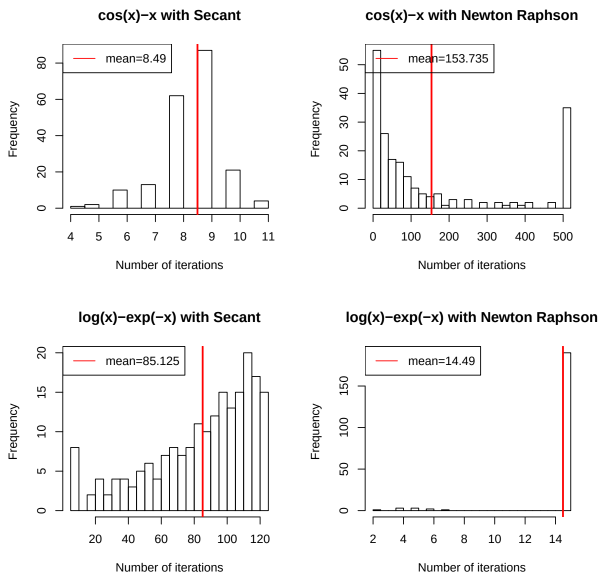
Figure 2 demonstrated the comparison between these two methods on number of iterations and final estimated roots using the 200 different initialization values produced. From figure (a), we can see that when applied to function $cos(x) - x$, which is essentially composed of a simple straightline function x and periodical curvy noise function $cos(x)$, Newton-Raphson takes much longer to converge or even fail to converge many times (mean iteration = 153.735), whereas secant is much faster to converge (mean converge = 8.49). When applied to function $log(x) - exp(-x)$, which has no periodical curvy noises, Newton-Raphson converges much faster (mean iteration = 14.49). From figure (b), we can see that for secant method increase the distance between the initialization value and the true solution does not affect too much number of iterations in function $cos(x) - x$, but does increase number of iterations in function $log(x) - exp(-x)$. Whereas for Newton-Raphson method, increase the distance between the initialization value and the true solution does affect number of iterations in function $cos(x) - x$, since it does not converge many of the times, but does not affect number of iterations in function $log(x) - exp(-x)$. From figure (c) and figure (d), we can observe similar patterns as in figure (a) and (b). While changing the initialization value makes the estimated root of $cos(x) - x$ differs a lot from the true solution for Newton-Raphson, it does not affect too much for secant method. On the other hand, in function $log(x) - exp(-x)$, Newton-Raphson gives more stable prediction of estimated root than secant method. Therefore, we conclude that secant method is legitimate for estimating the root of $cos(x) - x$ and that Newton-Raphson is better in estimating the root of $log(x) - exp(-x)$.

3.2.2 number of iterations and final estimated roots

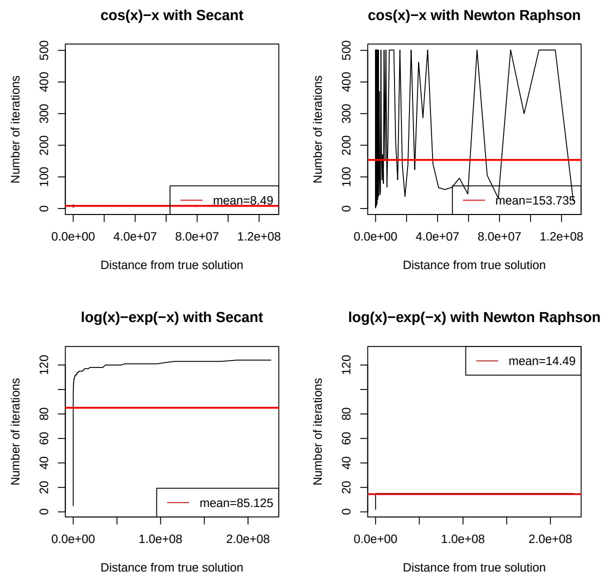
Figure 3 demonstrated the comparison between these two methods on speed using the 200 different initialization values produced. We can see that in estimating $cos(x) - x$ function, since Newton-Raphson takes longer time to converge, it is slower than secant method. Whereas for $log(x) - exp(-x)$, in which Newton-Raphson is faster to converge, it is faster than secant method.

4 Conclusion

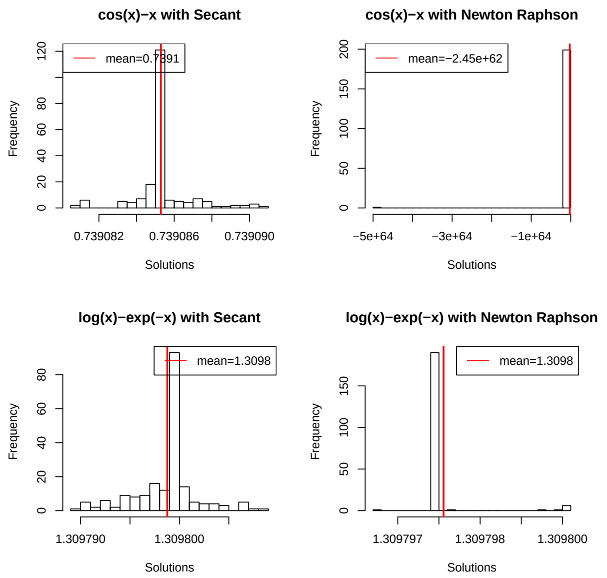
Here we applied Newton-Raphson and secant methods to estimate the roots of $cos(x) - x$ and $log(x) - exp(-x)$ and investigated how different initialization value impacts the number of iterations, speed and final estimated roots of these two different methods. We found that both methods are legitimate for estimating the root at different scenarios. Secant method is legitimate for estimating the root of $cos(x) - x$, which is essentially composed of a simple straightline function x and periodical curvy noise function $cos(x)$ and makes Newton-Raphson method fails to converge at different initialization values. Newton-Raphson is better in estimating the root of $log(x) - exp(-x)$, in terms of estimation accuracy and speed.



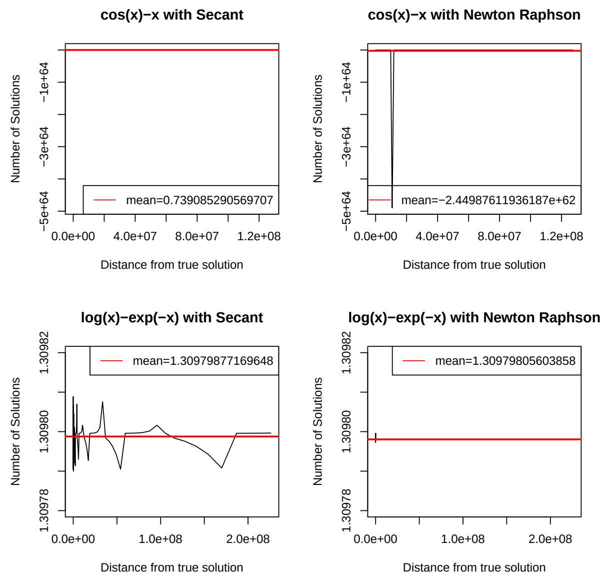
(a) Distribution of number of iterations



(b) Variation of number of iterations on distance between distance between initialization value with the true solution

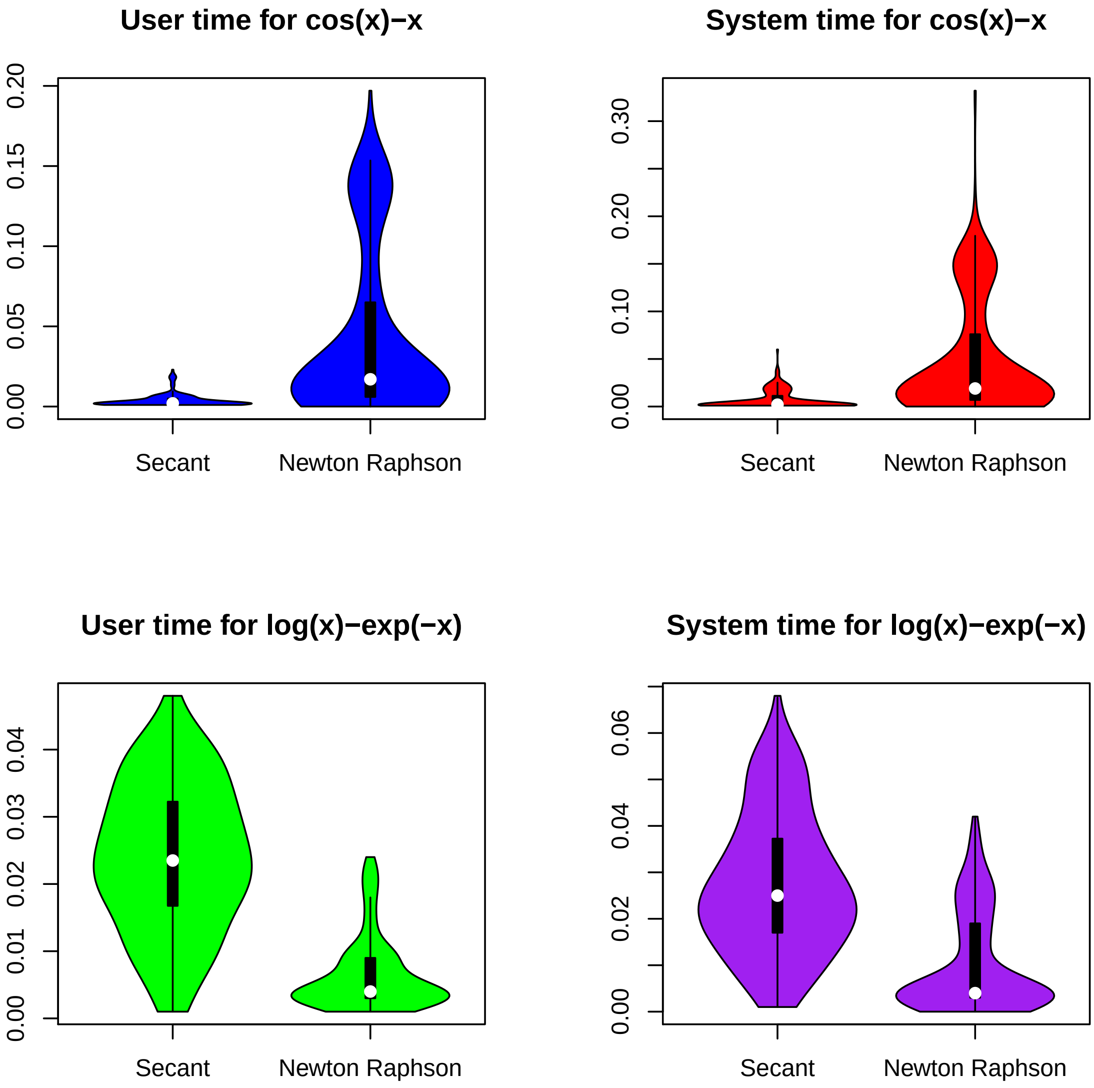


(c) Distribution of solutions



(d) Variation of solutions on distance between distance between initialization value with the true solution

Figure 2: The effect of different initialization values on number of iterations (to achieve convergence) and final estimated roots



(a) Distribution of time consumptions

Figure 3: The effect of different initialization values on speed