ex1

id:30788595-4

function

```
bisection<- function(fun=fun, digit = 10,a,b, step = F)
{
  if ((fun(a)>=0 & fun(b)>=0) || (fun(a)<0 & fun(b)<0) ) {
    print("A and B have an equal sign")
    return()
  }
  while ((b-a)>(10^-digit)) {
    c <- (a+b)/2
     if (step) {
      print(c)
    }
    if (fun(c)==0) {
      return(c)
    }else if (fun(a)*fun(c)>0) {
      a <- c
    }else {
      b <- c
    }
  }
  return(c)

}

Newton <- function(fun=fun, digit = 10,x, step = F){
  x_1 <- fun(x)
  while (round(x,digits = digit)!=round(x_1,digits = digit)) {
    if (step) {
      print(x_1)
    }
    x <- x_1
    x_1<- fun(x_1)
  }
return(x_1)
}
```

Q1

```
library(tictoc)
tic()
f_1 <- function(x){2*sin(x)-x}
bisection(fun = f_1,digit = 6,0.5*pi,pi,step = F)
```

```
## [1] 1.895495
```

```r
toc()
```

```
## 0.05 sec elapsed
```

```r
tic()
f_2 <- function(x){x - (2*sin(x)-x)/(2*cos(x)-1)}
Newton(fun = f_2,digit = 6,x = 0.5*pi,step = F)
```

```
## [1] 1.895494
```

```r
toc()
```

```
## 0 sec elapsed
```

The Newton method makes fewer iterations, as expected, but is not always faster. In each run, different results are obtained, and sometimes it is turns out that the bisection method is faster. The result is likely to come from R, and if we write in C or a more basic programming language we will get results that match that logic.

Q2

```r
tic()
f <- function(x) {x*(2-sqrt(2)*x)}
Newton(fun = f,digit = 6,x = 1)
```

```
## [1] 0.7071068
```

```r
toc()
```

```
## 0 sec elapsed
```

$$
\begin{gathered}
f(x) = 1/x - \sqrt{2} \\
f'(x) = -/x^2 \\
x_1 = x_0 - \frac{-1/x_0 - \sqrt{2}}{-1/x^2} \\
x_0 + x_0^2(1/x_0 - \sqrt{2}) \\
2x - \sqrt{2}x^2 \\
x_1 = x_0(2 - \sqrt{2}x_0)
\end{gathered}
$$

Because the function has one root, if the algorithm converged to a solution, it converged to the only solution.

Q3

```r
tic()
f_1 <- function(x){x - 1*((exp(x)-x-1)/(exp(x)-1))}
Newton(fun = f_1,digit = 6,x = 1)
```

```
## [1] 6.78183e-07
```

```r
toc()
```

```
## 0 sec elapsed
```

```r
tic()
f_2 <- function(x){x - 2*((exp(x)-x-1)/(exp(x)-1))}
Newton(fun = f_2,digit = 6,x = 1)
```

```
## [1] 1.086453e-11
```

```r
toc()
```

```
## 0.02 sec elapsed
```

The derivative is equal to: $e^x - 1 = 0$ It is easy to see that there is a single root at X = 0

Q4

```r
f <- function(x){x - atan(x)*((x^2)+1)}
Newton(fun = f,digit = 6,x = -1)
```

```
## [1] 0
```

$X_{n+1} = X_n - (1 + X_n^2) arctan(x_n)$ The Newton formula is used to find a local minimum. If X_0 is selected too large, you can see that X_{n+1} will be larger and the formula will start to explode. The way to find the solution is to place X_n and X_{n+1} the $X*$. We will receive: $f(X*) = 2x * -(1 + x*^2) arctan(x*)$ It is not hard to estimate \$x* = 1.3917 \$.