Noura mohamed ali code:5190832

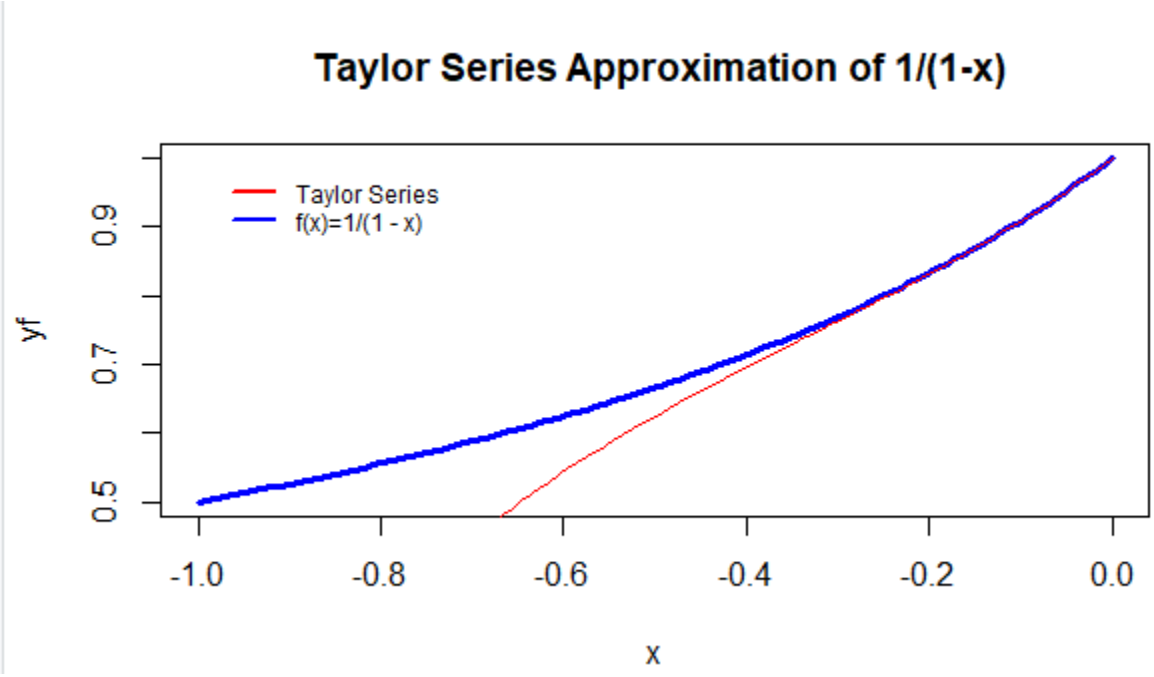
Q1

```
> ### 1)Newton method ########
> tol<-0.001
> initial<- 0
> f<-function(x){ 230*x^4+18*x^3+9*x^2-221*x-9}
> f(4)
[1] 59283
> fPrime <- function(x) {920*x^3+54*x^2+18*x-221}
> root <- function(f, fPrime, initial, tol) {
+   x = initial
+   while (abs(f(x)) > tol) {
+     x = x - f(x)/fPrime(x)
+   }
+   x
+ }
> root(f, fPrime, initial, tol)
[1] -0.04065929
```

Q2

```
> library(pracma)
> equation1 <- function(x) {1/(1-x)}
> p <- taylor(equation1, x0 = 0, n = 3)
> p
[1] 1.000003 1.000000 1.000000 1.000000
>
```

```
> ###Graphing Taylor Polynomial
> x <- seq(-1.0, 0, length.out=100)
> f <- function(x) (1/(1-x))
> p <- taylor(f, 0, 3)
> yf <- f(x)
> yp <- polyval(p, x)
> x
  [1] -1.00000000 -0.98989899 -0.97979798 -0.96969697 -0.95959596 -0.94949495
  [7] -0.93939394 -0.92929293 -0.91919192 -0.90909091 -0.89898990 -0.88888889
 [13] -0.87878788 -0.86868687 -0.85858586 -0.84848485 -0.83838384 -0.82828283
 [19] -0.81818182 -0.80808081 -0.79797980 -0.78787879 -0.77777778 -0.76767677
 [25] -0.75757576 -0.74747475 -0.73737374 -0.72727273 -0.71717172 -0.70707071
 [31] -0.69696970 -0.68686869 -0.67676768 -0.66666667 -0.65656566 -0.64646465
 [37] -0.63636364 -0.62626263 -0.61616162 -0.60606061 -0.59595960 -0.58585859
 [43] -0.57575758 -0.56565657 -0.55555556 -0.54545455 -0.53535354 -0.52525253
 [49] -0.51515152 -0.50505051 -0.49494949 -0.48484848 -0.47474747 -0.46464646
 [55] -0.45454545 -0.44444444 -0.43434343 -0.42424242 -0.41414141 -0.40404040
 [61] -0.39393939 -0.38383838 -0.37373737 -0.36363636 -0.35353535 -0.34343434
 [67] -0.33333333 -0.32323232 -0.31313131 -0.30303030 -0.29292929 -0.28282828
 [73] -0.27272727 -0.26262626 -0.25252525 -0.24242424 -0.23232323 -0.22222222
 [79] -0.21212121 -0.20202020 -0.19191919 -0.18181818 -0.17171717 -0.16161616
 [85] -0.15151515 -0.14141414 -0.13131313 -0.12121212 -0.11111111 -0.10101010
 [91] -0.09090909 -0.08080808 -0.07070707 -0.06060606 -0.05050505 -0.04040404
 [97] -0.03030303 -0.02020202 -0.01010101  0.00000000
> plot(x, yf, type = "l", main =' Taylor Series Approximation of 1/(1-x)',col
+      = "blue", lwd = 3)
> lines(x, yp, col = "red")
> legend('topleft', inset=.05, legend= c("Taylor Series", "f(x)=1/(1 - x)")
+        , lwd=c(2.5,2.5), col=c('red', 'blue'), bty='n', cex=.75)
```

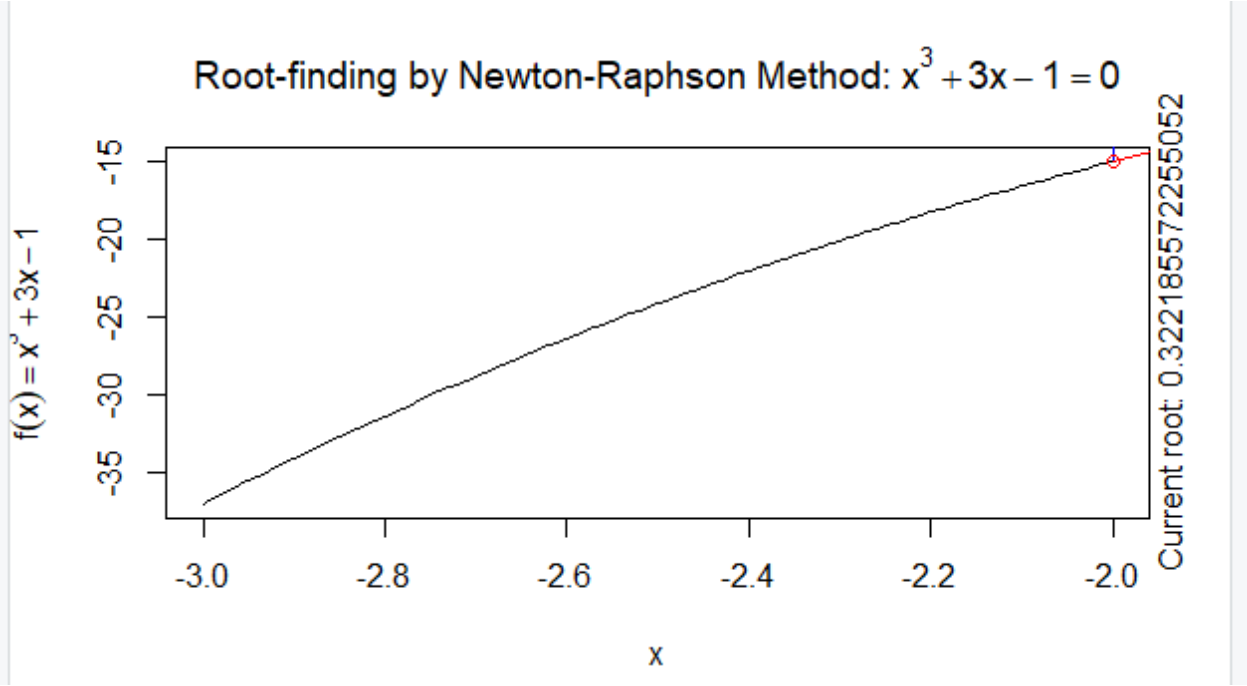


Q3

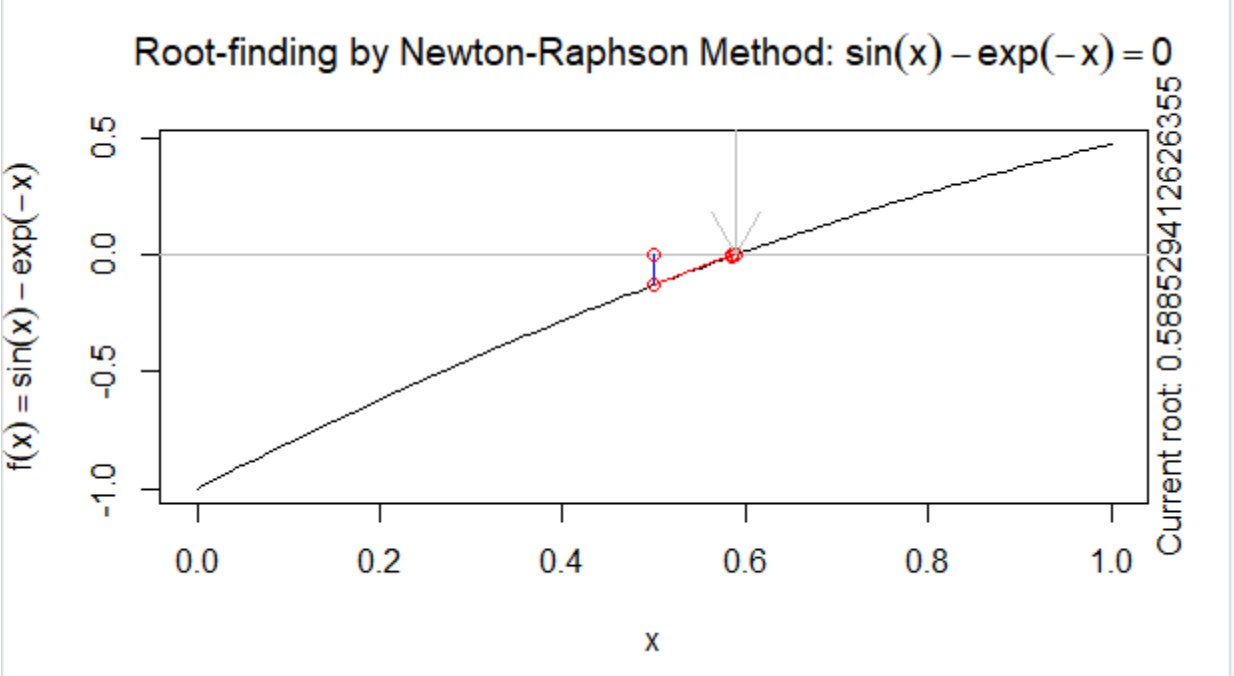
```
> ### 3) Bisection method --
> a<-1
> b<-2
> f<-function(x){x^3+4*x^2-10}
> f(a)
[1] -5
> f(b)  #conditions are satisfied
[1] 14
> #then calculate the midpoint
> c<-(a+b)/2
> while(f(c)!=0 && b-a >.02)
+ {
+   if(f(c)==0)
+   {
+     c
+   }
+   if(f(c)<0)
+   {
+     a<-c
+   }
+   else
+   {
+     b=c
+   }
+   {
+     c<-(a+b)/2;c
+   }
+ }
> c
[1] 1.367188
```

Q4

```
> ### 4)Newton method ########
> #a
> tol<-0.0001
> initial<- -2
> f<-function(x){ x^3+3*x-1}
> f(4)
[1] 75
> fPrime <- function(x) {3*x^2+3}
> root <- function(f, fPrime, initial, tol) {
+   x = initial
+   while (abs(f(x)) > tol) {
+     x = x - f(x)/fPrime(x)
+   }
+   x
+ }
> root(f, fPrime, initial, tol)
[1] 0.3221856
> newton.method(FUN = function(x) x^3 + 3*x -1, init = -2, rg
+               = c(-3, -2), tol = 0.0001)
>
```




```
> #b
> tol<-0.0001
> initial<- 0.5
> f<-function(x){ sin(x) - exp(-x)}
> f(4)
[1] -0.7751181
> fPrime <- function(x) {cos(x) + exp(-x)}
> root <- function(f, fPrime, initial, tol) {
+   x = initial
+   while (abs(f(x)) > tol) {
+     x = x - f(x)/fPrime(x)
+   }
+   x
+ }
> root(f, fPrime, initial, tol)
[1] 0.5885294
> newton.method(FUN = function(x) sin(x) - exp(-x), init = 0.5, rg
+               = c(0, 1), tol = 0.0001)
>
```



Q5

```
> ### 5) Fixed iteration method #########
> tol<-0.0001
> initial<- 1
> g<-function(x){(1/12)*(1+x^3)}
> root <- function(g, initial, tol) {
+   x = initial
+   while (abs(x-g(x)) > tol) {
+     x = g(x)
+   }
+   x
+ }
> root(g, initial, tol)
[1] 0.08338223
```

Q6

```
> #6) in trapezoidal we need to define the function,a,b,h=b-a/n
> f=function(x){y=x/((x^2+4))
+ return(y)}
> trapezoid= function(f, a, b, n) {
+
+   h <- 0.25
+
+   j <- 1:(n - 1)
+
+   xj <- a + j * h
+
+   value <- (h / 2) * (f(a) + 2 * sum(f(xj)) + f(b))
+
+   return(value)
+ }
> trapezoid(f,1,3,8)
[1] 0.4769769
> simpson <- function(f, a, b, n) {
+   h <- 0.25
+   x <- seq(a, b, h)
+
+   s <- f(x[1]) + f(x[n+1]) + 4*sum(f(x[seq(2,n,2)])) + 2 *sum(f(x[seq(3,n-1, 2)]))
+
+   s <- s*h/3
+   return(s)
+ }
> simpson(f,1,3,8)
[1] 0.4777546
```