Newton-Raphson Method

Yinqi Zhao

2/23/2020

A function which executes the Newto-Raphson method

```
newton <- function(fx, fx2, start, tol = 1e-3, maxitr = 1000){
  res <- start
  distance <- -Inf
  x1 <- start
  i <- 1
  while(i <= maxitr & abs(distance) > tol){
    x2 <- x1 - fx(x1) / fx2(x1)
    res <- c(res, x2)
    distance <- abs(x2 - x1)
    x1 <- x2
    print(paste0("iteration ", i, ": ", "X = ", x1, ", Y = ", fx(x1)))
    i <- i + 1
  }
  if(abs(distance) > tol){
    print("fail to converge, please try another start value")
  }
  print(paste0("converged! the root is ", x1))
  return(list(root = x2,
              res = res))
}
```

A function for plotting the process of the Newton-Raphson method

```
plot.process <- function(fx, fx2, res, xrange = c(-1, 1)){
  grid <- seq(from = xrange[1], to = xrange[2], length.out = 1000)
  plot(grid, fx(grid), type = "l", xlab = "x", ylab = "y")
  abline(a = 0, b = 0)
  for(i in 1:length(res)){
    x <- res[i]
    segments(x0 = x, y0 = 0, x1 = x, y1 = fx(x), col = "red")
    if(i > 1){
      segments(x0 = x, y0 = 0, x1 = res[i - 1], y1 = fx(res[i - 1]), col = "red", lty = 5)
    }
  }
}
```

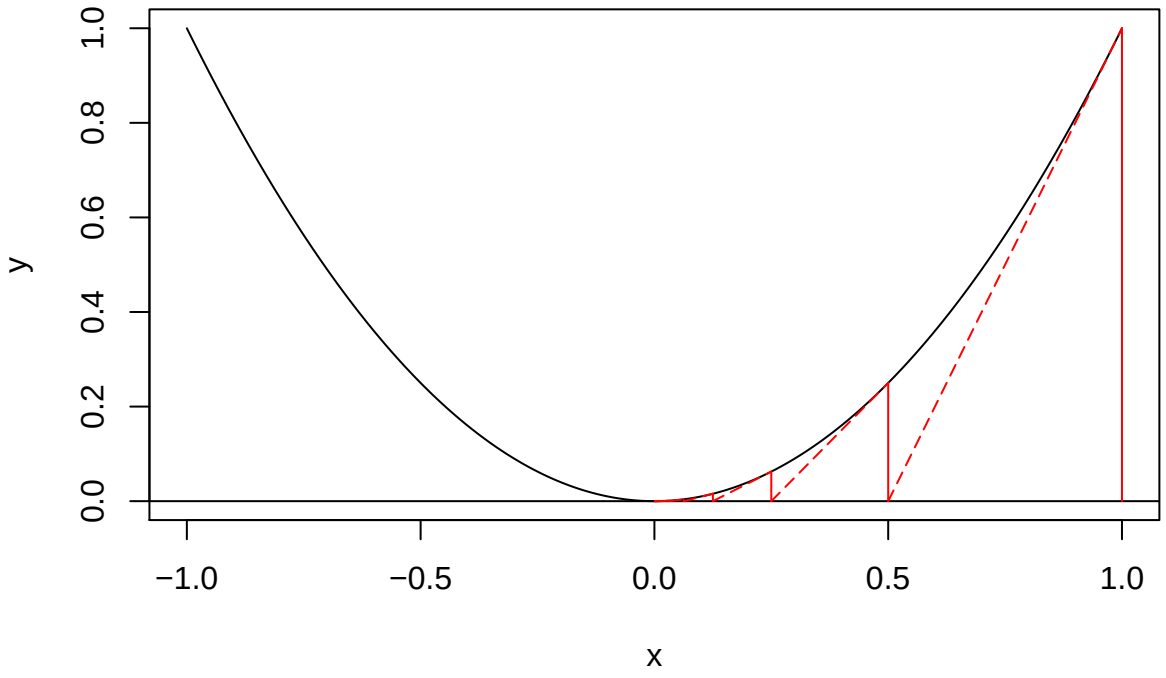
1. $f(x) = x^2$

```
fx.1 <- function(x){
  return(x^2)
}
fx2.1 <- function(x){
  return(2 * x)
}

temp <- newton(fx = fx.1, fx2 = fx2.1, start = 1)
```

```
## [1] "iteration 1: X = 0.5, Y = 0.25"
## [1] "iteration 2: X = 0.25, Y = 0.0625"
## [1] "iteration 3: X = 0.125, Y = 0.015625"
## [1] "iteration 4: X = 0.0625, Y = 0.00390625"
## [1] "iteration 5: X = 0.03125, Y = 0.0009765625"
## [1] "iteration 6: X = 0.015625, Y = 0.000244140625"
## [1] "iteration 7: X = 0.0078125, Y = 6.103515625e-05"
## [1] "iteration 8: X = 0.00390625, Y = 1.52587890625e-05"
## [1] "iteration 9: X = 0.001953125, Y = 3.814697265625e-06"
## [1] "iteration 10: X = 0.0009765625, Y = 9.5367431640625e-07"
## [1] "converged! the root is 0.0009765625"
```

```
plot.process(fx.1, fx.2, res = temp$res)
```



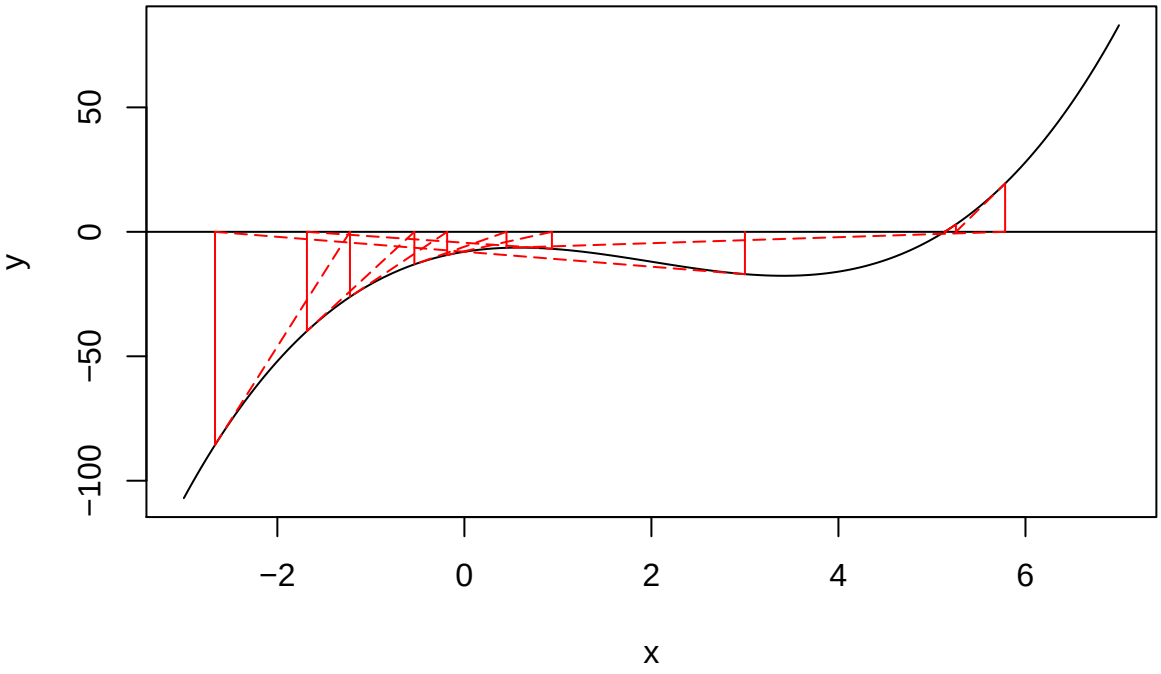
2. $f(x) = (x-2)^3 - 6x$

```
fx.1 <- function(x){
  return((x - 2)^3 - 6 * x)
}
fx2.1 <- function(x){
  return(3 * (x - 2)^2 - 6)
}
```

```
temp <- newton(fx = fx.1, fx2 = fx2.1, start = 3)
```

```
## [1] "iteration 1: X = -2.66666666666667, Y = -85.6296296296296"
## [1] "iteration 2: X = -1.22347066167291, Y = -26.1534960566936"
## [1] "iteration 3: X = -0.184491023221188, Y = -9.31744727467655"
## [1] "iteration 4: X = 0.935932715447937, Y = -6.82037496885554"
## [1] "iteration 5: X = -1.68398084415805, Y = -39.8940525021216"
## [1] "iteration 6: X = -0.534797887141653, Y = -13.0777968946937"
## [1] "iteration 7: X = 0.450302290885039, Y = -6.42351040864467"
## [1] "iteration 8: X = 5.78239262658787, Y = 19.418421589051"
## [1] "iteration 9: X = 5.25642583165884, Y = 2.99359140636451"
## [1] "iteration 10: X = 5.14045326730023, Y = 0.129833434462519"
## [1] "iteration 11: X = 5.13494889785823, Y = 0.000285282368981399"
## [1] "iteration 12: X = 5.13493674976395, Y = 1.38793154746963e-09"
## [1] "converged! the root is 5.13493674976395"
```

```
plot.process(fx.1, fx.2, res = temp$res, xrange = c(-3, 7))
```



3. $f(x) = \sin x$

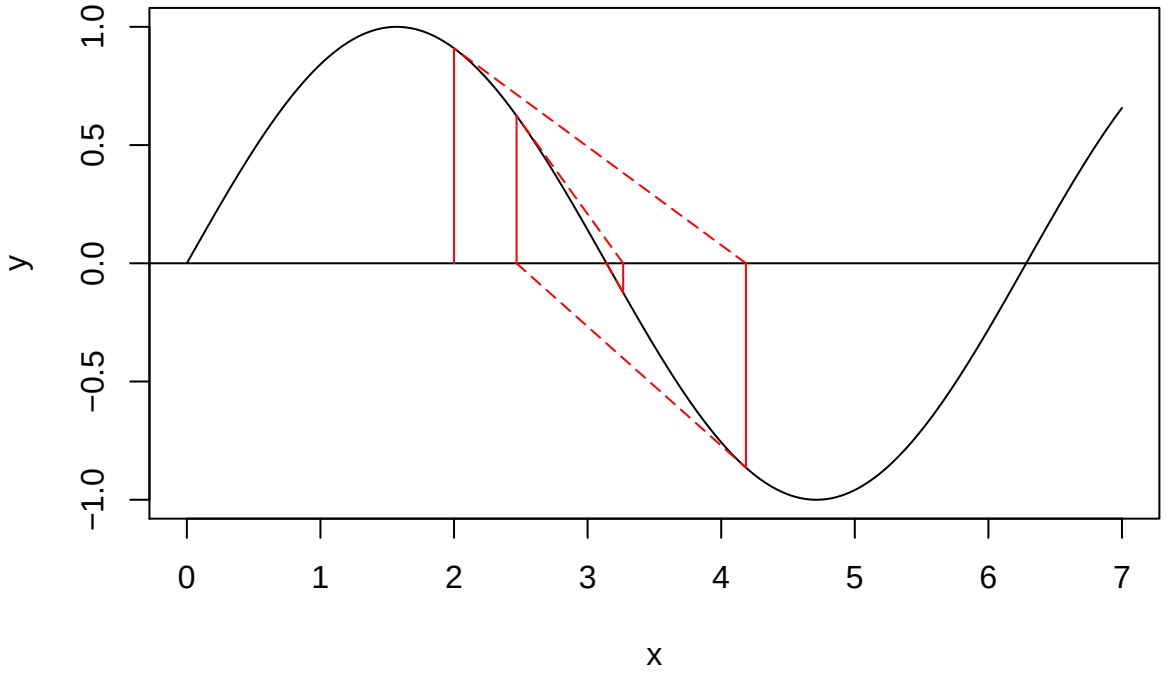
```
fx.1 <- function(x){
  return(sin(x))
}
fx2.1 <- function(x){
  return(cos(x))
}

temp <- newton(fx = fx.1, fx2 = fx2.1, start = 2)
```

```
## [1] "iteration 1: X = 4.18503986326152, Y = -0.864144147074565"
## [1] "iteration 2: X = 2.46789367451467, Y = 0.623881072066777"
## [1] "iteration 3: X = 3.26618627756911, Y = -0.124271517762097"
```

```
## [1] "iteration 4: X = 3.14094391231764, Y = 0.000648741226652542"
## [1] "iteration 5: X = 3.1415926536808, Y = -9.10110761127821e-11"
## [1] "converged! the root is 3.1415926536808"
```

```
plot.process(fx.1, fx.2, res = temp$res, xrange = c(0, 7))
```



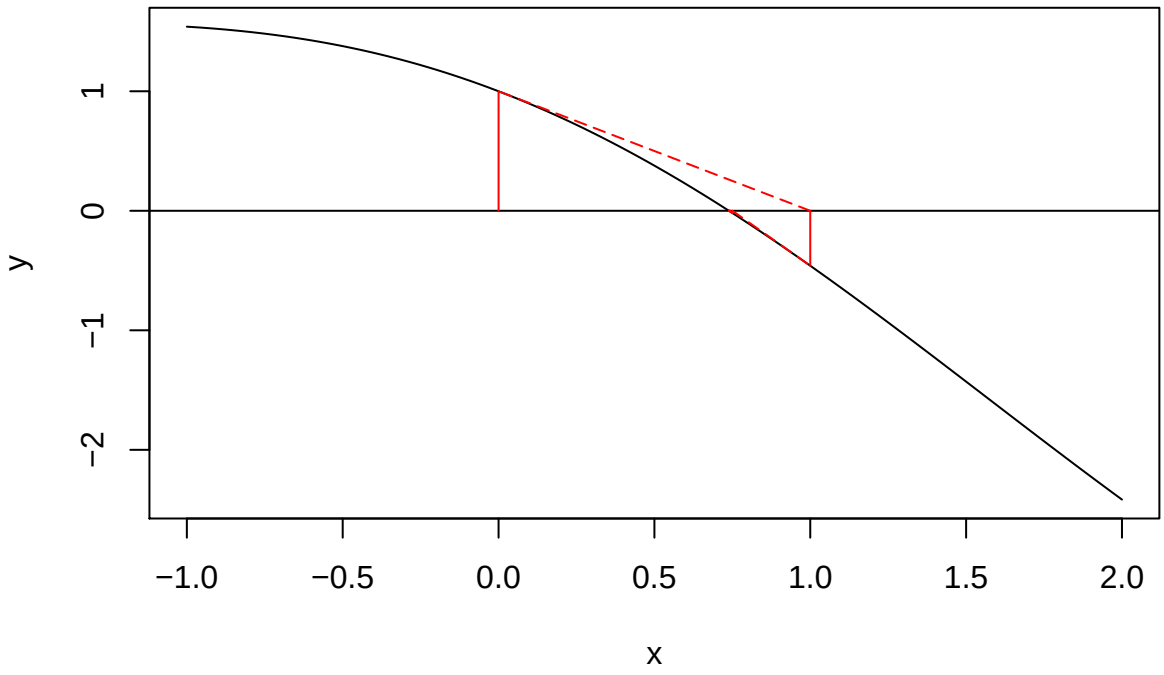
4. $f(x) = \cos x - x$

```
fx.1 <- function(x){
  return(cos(x) - x)
}
fx2.1 <- function(x){
  return(-sin(x) - 1)
}

temp <- newton(fx = fx.1, fx2 = fx2.1, start = 0)
```

```
## [1] "iteration 1: X = 1, Y = -0.45969769413186"
## [1] "iteration 2: X = 0.750363867840244, Y = -0.0189230738221174"
## [1] "iteration 3: X = 0.739112890911362, Y = -4.64558989908825e-05"
## [1] "iteration 4: X = 0.739085133385284, Y = -2.84720469423405e-10"
## [1] "converged! the root is 0.739085133385284"
```

```
plot.process(fx.1, fx.2, res = temp$res, xrange = c(-1, 2))
```


y
1
0
−1
−2
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x