MA588

R-Programming Lab

Lab 3

SAGAR KUMAR

194161013

Data Science

Question1

Find the minimum of the function f(x)= `x1-x2+2*x1^2+2*x1*x2+x2^2` using Newton's method.

Libraries

```
library(numDeriv)
library(pracma)
```

Defining function

f(x)

```
func<- function(x) {
  x1<-x[1]
  x2<-x[2]

  result<-x1-x2+2*x1^2+2*x1*x2+x2^2
  result
}
```

Hessian Matrix of f(x)

```
hessian_mat=hessian(func,c(0,0))
hessian_matrix<-as.matrix(hessian_mat)
cat("Hessian of function f(x) is:")

## Hessian of function f(x) is:

hessian_matrix

##      [,1] [,2]
## [1,]    4    2
## [2,]    2    2
```

Inverse of hessian matrix

```
H=inv(hessian_matrix)   #Inverse of hessian matrix
H

##      [,1] [,2]
## [1,]  0.5 -0.5
## [2,] -0.5  1.0
```

tolerance and iteration

```
tolerance <- c(0,0)  #set tolerance=0 (when gradient of function is zero
Then x is optimum point)
k=0  #Here k is iteration
```

code of newton method for finding minima of function

```
root<-function(x,g,tolerance){
  while( (all(abs(g)>=tolerance)) ){
    if(k <101){
      g=x
      x=x-H %*% grad(func,g)
      k=k+1
      #print(k)
      #print("\n")
      #print(x)

    }
    else{
      break;
    }




  }
  return(x)
}
```

choosing randomly 100 guess

```
a=seq(0,4.95,0.05)
b=seq(0,4.95,0.05)

data.frame(a,b)
```

```
##        a    b
## 1   0.00 0.00
## 2   0.05 0.05
## 3   0.10 0.10
## 4   0.15 0.15
## 5   0.20 0.20
## 6   0.25 0.25
## 7   0.30 0.30
## 8   0.35 0.35
## 9   0.40 0.40
## 10  0.45 0.45
## 11  0.50 0.50
## 12  0.55 0.55
## 13  0.60 0.60
## 14  0.65 0.65
## 15  0.70 0.70
## 16  0.75 0.75
## 17  0.80 0.80
## 18  0.85 0.85
```

```
## 19  0.90 0.90
## 20  0.95 0.95
## 21  1.00 1.00
## 22  1.05 1.05
## 23  1.10 1.10
## 24  1.15 1.15
## 25  1.20 1.20
## 26  1.25 1.25
## 27  1.30 1.30
## 28  1.35 1.35
## 29  1.40 1.40
## 30  1.45 1.45
## 31  1.50 1.50
## 32  1.55 1.55
## 33  1.60 1.60
## 34  1.65 1.65
## 35  1.70 1.70
## 36  1.75 1.75
## 37  1.80 1.80
## 38  1.85 1.85
## 39  1.90 1.90
## 40  1.95 1.95
## 41  2.00 2.00
## 42  2.05 2.05
## 43  2.10 2.10
## 44  2.15 2.15
## 45  2.20 2.20
## 46  2.25 2.25
## 47  2.30 2.30
## 48  2.35 2.35
## 49  2.40 2.40
## 50  2.45 2.45
## 51  2.50 2.50
## 52  2.55 2.55
## 53  2.60 2.60
## 54  2.65 2.65
## 55  2.70 2.70
## 56  2.75 2.75
## 57  2.80 2.80
## 58  2.85 2.85
## 59  2.90 2.90
## 60  2.95 2.95
## 61  3.00 3.00
## 62  3.05 3.05
## 63  3.10 3.10
## 64  3.15 3.15
## 65  3.20 3.20
## 66  3.25 3.25
## 67  3.30 3.30
## 68  3.35 3.35
```

```
## 69  3.40 3.40
## 70  3.45 3.45
## 71  3.50 3.50
## 72  3.55 3.55
## 73  3.60 3.60
## 74  3.65 3.65
## 75  3.70 3.70
## 76  3.75 3.75
## 77  3.80 3.80
## 78  3.85 3.85
## 79  3.90 3.90
## 80  3.95 3.95
## 81  4.00 4.00
## 82  4.05 4.05
## 83  4.10 4.10
## 84  4.15 4.15
## 85  4.20 4.20
## 86  4.25 4.25
## 87  4.30 4.30
## 88  4.35 4.35
## 89  4.40 4.40
## 90  4.45 4.45
## 91  4.50 4.50
## 92  4.55 4.55
## 93  4.60 4.60
## 94  4.65 4.65
## 95  4.70 4.70
## 96  4.75 4.75
## 97  4.80 4.80
## 98  4.85 4.85
## 99  4.90 4.90
## 100 4.95 4.95
```

Each row in dataframe represents initial value of x. And from above dataframe you can see that there are 100 differents points and point (0,0) is included at index 1

```
m<-matrix(,nrow=2,ncol=100)
```

Finding minima of function for 100 points

```
for(i in 1:100){
  x<-c(a[i],b[i])
  m[,i]=root(x,grad(func,x),tolerance)

}
```

minimum value of f(x)

```
m
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
[,14]
## [1,] -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  -1.0  -1.0  -1.0  -1.0
-1.0
## [2,]  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5   1.5   1.5   1.5   1.5
1.5
##      [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
[,26]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
[,38]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49]
[,50]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60] [,61]
[,62]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72] [,73]
[,74]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
```

```
1.5
##      [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84] [,85]
[,86]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96] [,97]
[,98]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,99] [,100]
## [1,]  -1.0   -1.0
## [2,]   1.5    1.5
```

from above matrix(2*100) you can observe that the Each column represents the minimum of function calculating using newton's method (implemented from scratch) for different starting values .
we are taken 100 such differnet values and minimum came out to be the same value for each case .

Value of function f(x) at mimima (x1=-1,x2=1.5)

```
ans<-0
for(i in 1:100){
   ans[i]<-func(c(m[,i]))

}

ans
```

```
##   [1] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [13] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [25] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [37] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [49] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [61] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [73] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [85] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [97] -1.25 -1.25 -1.25 -1.25
```

using inbuilt function(nleqslv)

Finding minima function using library function

```
f <- function(x){
  y <- numeric(2)
  y[1] <- 4*x[1] + 2*x[2]+1
  y[2] <-  2*x[1] + 2*x[2]- 1
  y
}

jacob <- function(x) {
  n <- length(x)
  Df <- matrix(numeric(n*n),n,n)
  Df[1,1] <- 4
  Df[1,2] <- 2
  Df[2,1] <- 2
  Df[2,2] <- 2
  Df
}

a=seq(-1,4.95,0.06)
b=seq(-1,4.95,0.06)
library(numDeriv)
library(pracma)

library(nleqslv)
mat<-matrix(,nrow=2,ncol=100)


for(i in 1:100){
  xstart <- c(a[i],b[i])
  fstart <- jacob(xstart)
  xstart
  fstart
  mat[,i]<-nleqslv(xstart,f,method="Newton")$x

}
```

minimum value of f(x)

```
mat

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
[,14]
## [1,] -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  -1.0  -1.0  -1.0  -1.0
-1.0
## [2,]  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5   1.5   1.5   1.5   1.5
1.5
##      [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
[,26]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
[,38]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49]
[,50]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60] [,61]
[,62]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72] [,73]
[,74]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84] [,85]
[,86]
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96] [,97]
[,98]
```

```
## [1,]  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -
1.0
## [2,]   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5   1.5
1.5
##      [,99] [,100]
## [1,]  -1.0   -1.0
## [2,]   1.5    1.5
```

from above matrix(2*100) you can observe that the Each column represents the minimum of function calculating using newton's method (implemented using inbuilt function) for different starting values .
we are taken 100 such differnet values and minimum came out to be the same value for each case .

Value of function f(x) at minima (using inbuilt function)

```
ans_inbuilt<-0
for(i in 1:100){
   ans_inbuilt[i]<-func(c(mat[,i]))

}

ans_inbuilt
```

```
##   [1] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [13] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [25] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [37] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [49] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [61] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [73] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [85] -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -1.25 -
1.25
##  [97] -1.25 -1.25 -1.25 -1.25
```

Solution Agreement with library function

```
own_ans<-ans
lib_ans<-ans_inbuilt

data.frame(own_ans,lib_ans)
```

```
##     own_ans lib_ans
## 1     -1.25   -1.25
## 2     -1.25   -1.25
## 3     -1.25   -1.25
## 4     -1.25   -1.25
## 5     -1.25   -1.25
## 6     -1.25   -1.25
## 7     -1.25   -1.25
## 8     -1.25   -1.25
## 9     -1.25   -1.25
## 10    -1.25   -1.25
## 11    -1.25   -1.25
## 12    -1.25   -1.25
## 13    -1.25   -1.25
## 14    -1.25   -1.25
## 15    -1.25   -1.25
## 16    -1.25   -1.25
## 17    -1.25   -1.25
## 18    -1.25   -1.25
## 19    -1.25   -1.25
## 20    -1.25   -1.25
## 21    -1.25   -1.25
## 22    -1.25   -1.25
## 23    -1.25   -1.25
## 24    -1.25   -1.25
## 25    -1.25   -1.25
## 26    -1.25   -1.25
## 27    -1.25   -1.25
## 28    -1.25   -1.25
## 29    -1.25   -1.25
## 30    -1.25   -1.25
## 31    -1.25   -1.25
## 32    -1.25   -1.25
## 33    -1.25   -1.25
## 34    -1.25   -1.25
## 35    -1.25   -1.25
## 36    -1.25   -1.25
## 37    -1.25   -1.25
## 38    -1.25   -1.25
## 39    -1.25   -1.25
## 40    -1.25   -1.25
## 41    -1.25   -1.25
```

```
## 42    -1.25   -1.25
## 43    -1.25   -1.25
## 44    -1.25   -1.25
## 45    -1.25   -1.25
## 46    -1.25   -1.25
## 47    -1.25   -1.25
## 48    -1.25   -1.25
## 49    -1.25   -1.25
## 50    -1.25   -1.25
## 51    -1.25   -1.25
## 52    -1.25   -1.25
## 53    -1.25   -1.25
## 54    -1.25   -1.25
## 55    -1.25   -1.25
## 56    -1.25   -1.25
## 57    -1.25   -1.25
## 58    -1.25   -1.25
## 59    -1.25   -1.25
## 60    -1.25   -1.25
## 61    -1.25   -1.25
## 62    -1.25   -1.25
## 63    -1.25   -1.25
## 64    -1.25   -1.25
## 65    -1.25   -1.25
## 66    -1.25   -1.25
## 67    -1.25   -1.25
## 68    -1.25   -1.25
## 69    -1.25   -1.25
## 70    -1.25   -1.25
## 71    -1.25   -1.25
## 72    -1.25   -1.25
## 73    -1.25   -1.25
## 74    -1.25   -1.25
## 75    -1.25   -1.25
## 76    -1.25   -1.25
## 77    -1.25   -1.25
## 78    -1.25   -1.25
## 79    -1.25   -1.25
## 80    -1.25   -1.25
## 81    -1.25   -1.25
## 82    -1.25   -1.25
## 83    -1.25   -1.25
## 84    -1.25   -1.25
## 85    -1.25   -1.25
## 86    -1.25   -1.25
## 87    -1.25   -1.25
## 88    -1.25   -1.25
## 89    -1.25   -1.25
## 90    -1.25   -1.25
## 91    -1.25   -1.25
```

```
## 92    -1.25   -1.25
## 93    -1.25   -1.25
## 94    -1.25   -1.25
## 95    -1.25   -1.25
## 96    -1.25   -1.25
## 97    -1.25   -1.25
## 98    -1.25   -1.25
## 99    -1.25   -1.25
## 100   -1.25   -1.25
```

from above dataframe you can observe that the data in column 1(own_ans) is exactly equal to column 2(lib_ans) as expected.