732A90 Lab 2

Daniele Bozzi & Hugo Morvan

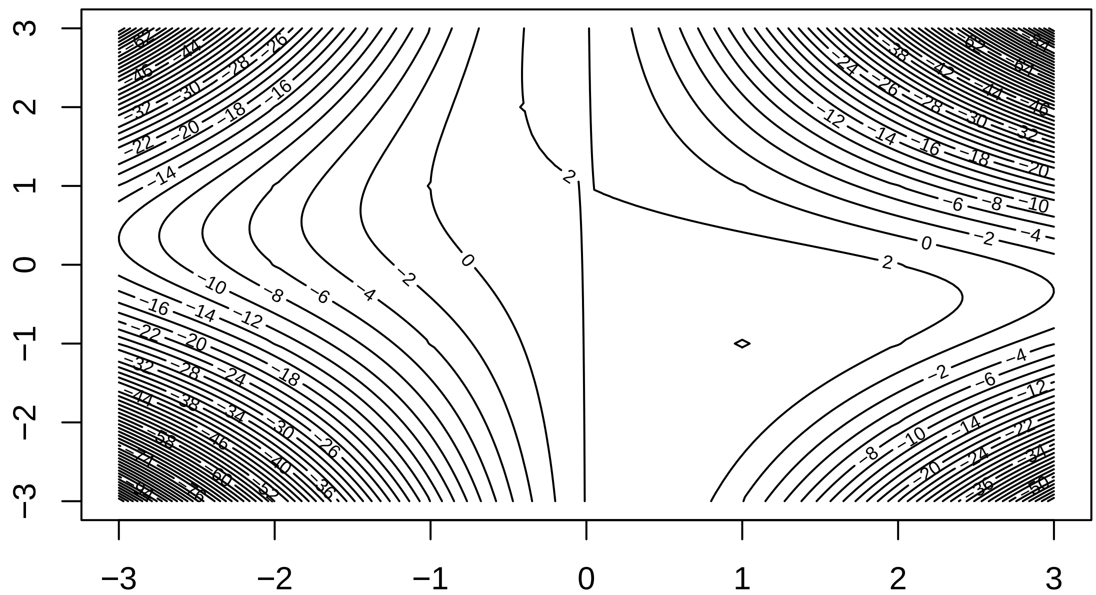
2023-11-08

Question 1: Optimisation of a two-dimensional function

a)

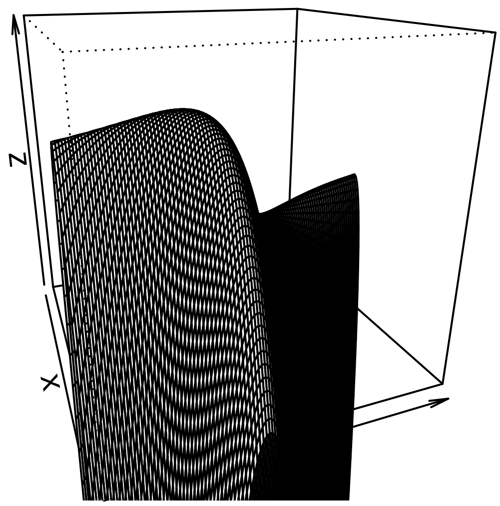
```r
f <- function(p){
  x <- p[1]
  y <- p[2]
  return(-x^2 -x^2*y^2 -2*x*y +2*x +2)
}
f_x <- function(p){
  x <- p[1]
  y <- p[2]
  return(-2*x -2*x*y^2 -2*y +2)
}
f_y <- function(p){
  x <- p[1]
  y <- p[2]
  return(-2*x^2*y-2*x)
}
f_xy <- function(p){
  x <- p[1]
  y <- p[2]
  return(-4*x*y-2)
}
f_xx <- function(p){
  x <- p[1]
  y <- p[2]
  return(-2 -2*y^2)
}
f_yy <- function(p){
  x <- p[1]
  y <- p[2]
  return(-2*x^2)
}
get_gradient <- function(p){
  return(c(f_x(p), f_y(p)))
}
get_hessian <- function(p){
  hess <- matrix(c(f_xx(p), f_xy(p), f_xy(p), f_yy(p)), ncol=2)
  return(hess)
}

x <- seq(-3,3,0.05)
```

```
y <- seq(-3,3,0.05)

lx <- length(x)
ly <- length(y)

points1 <- matrix(0,nrow=lx, ncol=ly)

for (i in 1:lx){
  for (q in 1:ly){
    points1[i,q] <- f(c(x[i], y[q]))
  }
}

contour(x,y,points1, nlevels=50)
```




```
# unnecessary not so good looking 3d plot
persp(x, y, points1, xlab="x", ylab="y", zlab="z", theta=69, phi=20, zlim=c(0, 5))
```

```
## Warning in persp.default(x, y, points1, xlab = "x", ylab = "y", zlab = "z", :
## la surface s'étend au delà de la boite
```



b)

```
newton <- function(x, epsilon){
  #x is you starting vector
  dist <- 999

  while(dist > epsilon){
    gp <- get_gradient(x)
    gpp <- get_hessian(x)
    x_next <- x - solve(gpp)%*%gp
    dist <- sum((x-x_next)*(x-x_next))
    x <- x_next
  }
  return(x)
}
```

c)

```
p1 <- c(2,0)
p2 <- c(-1,-2)
p3 <- c(0,1)
p4 <- c(10,-10)
epsilon <- 1E-8
newton(p1, epsilon)
```

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

```r
newton(p2, epsilon)
```

```
##              [,1]
## [1,] 1.361391e-22
## [2,] 1.000000e+00
```

```r
newton(p3, epsilon)
```

```
##      [,1]
## [1,]    0
## [2,]    1
```

```r
newton(p4, epsilon)
```

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

We obtain 2 points : (0,1) and (1,-1)

```r
print("Gradient and Hessian for the point (0,1)")
```

```
## [1] "Gradient and Hessian for the point (0,1)"
```

```r
get_gradient(c(0,1))
```

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

```r
get_hessian(c(0,1))
```

```
##      [,1] [,2]
## [1,]   -4   -2
## [2,]   -2    0
```

```r
print("this is a saddle point")
```

```
## [1] "this is a saddle point"
```

```r
#print("the matrix is negative semi-definite")

print("Gradient and Hessian for the point (1,-1)")
```

```
## [1] "Gradient and Hessian for the point (1,-1)"
```

```r
get_gradient(c(1,-1))
```

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

```r
get_hessian(c(1,-1))
```

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

```r
print("this is a maximum")
```

```
## [1] "this is a maximum"
```

d)

Question 2:

a)

b)

c)

d)