Homework 8 :: MATH 504 :: Due Tuesday, Nov 1st, 11:59 pm

Your homework submission must be a single pdf called "LASTNAME-hw7.pdf" with your solutions to all theory problem to receive full credit. All answers must be typed in Latex.

1. Let $f(x) = -x_1^2 - 4x_2^2$. Consider two different points

$$\tilde{x} = [2, 0]^T \qquad \textit{and} \qquad \bar{x} = [\sqrt{3}, 1/2]^T$$

Show that

$$\nabla f(\tilde{x})^T x'(\tilde{t}) = \nabla f(\bar{x})^T x'(\bar{t}) = 0.$$

Hint: Consider the level set of f at the level $c = -4$. Define a parametric curve of a curve passing through $\tilde{x}$ and $\bar{x}$, similar to what we did in lecture.

Solution: The level set at $c = -4$ consists of the following points:

$$\{x, y \in \mathbb{R}^{\nvDash} | -x^2 - 4y^2 = -4\} = \{x, y \in \mathbb{R}^{\nvDash} | x^2/4 + y^2 = 1\},$$

which is the equation of an ellipse, which passes through the points $\tilde{x}$ and $\bar{x}$. Then, the parametric form of the equation can be expressed as follows:

$$\gamma = \{x(t) = [2\sqrt{1-t^2}, t]^T : t \in (-1, 1)\}.$$

The curve passes through $\tilde{x}$ at $t = 0$ and through $\bar{x}$ at $t = 1/2$. Then,

$$x'(t) = \left[\frac{-2t}{\sqrt{1-t^2}}, 1\right]^T,$$

implying that $x'(t = 0) = [0, 1]^T$ and $x'(t = 1/2) = [-2/\sqrt{3}, 1]^T$. Next, we have that

$$\nabla f = [-2x, -8y]^T \Rightarrow \nabla f(\tilde{x}) = [-4, 0] \quad \text{and} \quad \nabla f(\bar{x}) = [-2\sqrt{3}, -4].$$

Finally, we have

$$\nabla^T f(\tilde{x}) x'(t = 0) = [-4, 0][0, 1]^T = 0$$
$$\nabla^T f(\bar{x}) x'(t = 1/2) = [-2\sqrt{3}, -4][-2/\sqrt{3}, 1]^T = 4 - 4 = 0.$$

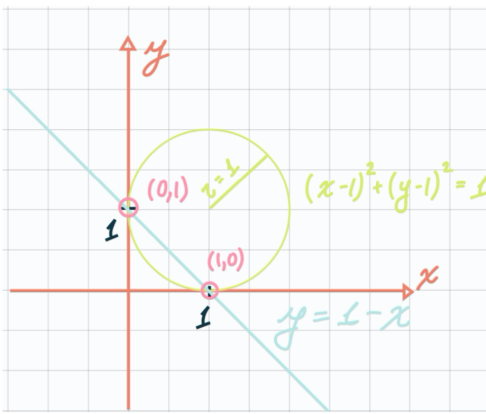
2. Consider the system of equations

$$(x-1)^2 + (y-1)^2 - 1 = 0$$
$$x + y - 1 = 0$$

a. Draw the set of points on the plane that satisfy each equation, and indicate the solutions of the system.

Solution: The first equation constitutes a circle centered at $(1, 1)$ with a radius of 1. Thus, the set of all points constituting the green circle below satisfy the first equation. On the other hand, the second equation represents a line with a slope of -1 and y-intercept of 1. All the points on the light blue line satisfy the second equation. On the other hand, only $(0, 1)$ and $(1, 0)$, the interaction points of both graphs marked by pink circles, solve the system.

```
knitr::include_graphics('C:/Users/8levs/Downloads/SmartSelect_20220305-162356_Samsung Notes.jpg'
)
```



b. Solve the system exactly.

Solution: From the second equation we have that $x - 1 = -y$, which, when substituted in the first equation yields in the following:

$$(-y)^2 + (y-1)^2 = 1,$$
$$y^2 + y^2 - 2y + 1 = 1,$$
$$2(y^2 - y) = 0,$$
$$y(y-1) = 0,$$

meaning that y can be either 0 or 1. When $y = 0$, we have that $x - 1 = 0$, implying that $x = 1$. Similarly, when $y = 1$, we have that $x = -1 + 1 = 0$. Thus, the exact solutions are $\{(1, 0);\ (0, 1)\}$.

c. Apply Newton's method twice with $[x_0, y_0]^T = [1/2, 1/2]^T$. Illustrate the corresponding steps geometrically.

Solution: We first rewrite the system as follows:

$$F(x,y) = \begin{cases} f_1(x,y) = (x-1)^2 + (y-1)^2 - 1 = 0 \\ f_2(x,y) = y + x - 1 = 0. \end{cases}$$

Then, the Jacobian matrix is given by:

$$J_F(x,y) = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} 2(x-1) & 2(y-1) \\ 1 & 1 \end{bmatrix}.$$

Thus, when using Newton's method for $[x_0, y_0]^T = [1/2, 1/2]^T$, we have that:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} - \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -\frac{1}{2} \\ 0 \end{bmatrix}$$

.

However, we note that the Jacobian evaluated at our initial guess point is singular; namely,

$$\begin{vmatrix} -1 & -1 \\ 1 & 1 \end{vmatrix} = -1 - (-1) = 0.$$

Thus, we add some noise to the diagonal elements to avoid this situation.

```
J<-matrix(c(-1,-1,1,1),nrow=2, byrow = T)
alpha<-0.05
I<-matrix(c(1,0,0,1), nrow=2, byrow=T)
(newJ<-J+alpha*I)
```

```
##       [,1]  [,2]
## [1,] -0.95 -1.00
## [2,]  1.00  1.05
```

```
(invJ<-solve(newJ))
```

```
##      [,1] [,2]
## [1,]  420  400
## [2,] -400 -380
```

Using the new Jacobian matrix, we have:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} - \begin{bmatrix} 420 & 400 \\ -400 & -380 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} 210.5 \\ -199.5 \end{bmatrix}.$$

```
matrix(c(1/2,1/2), ncol=1)-invJ%*%matrix(c(-1/2,0), ncol=1)
```

```
##          [,1]
## [1,]  210.5
## [2,] -199.5
```

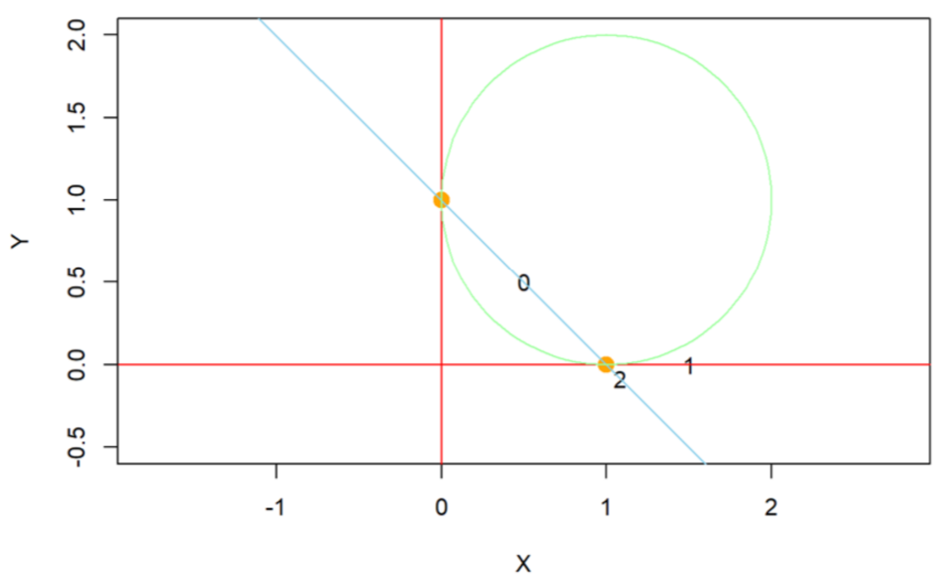
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 210.5 \\ -199.5 \end{bmatrix} - \begin{bmatrix} 2(210.5-1) & 2(-199.5-1) \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 83291.5 \\ 10 \end{bmatrix} = \begin{bmatrix} 104.0348 \\ -103.0348 \end{bmatrix}.$$

```
matrix(c(210.5, -199.5), ncol=1)-solve(matrix(c(2*(210.5-1), 2*(-199.5-1),1,1), ncol=2, byrow=T
))%*%(matrix(c(83291.5,10), ncol=1))
```

```
##             [,1]
## [1,]  104.0348
## [2,] -103.0348
```

Since using $\alpha = 0.05$ yielded in bad estimates for the solutions in the first two iterations, I will use $\alpha = 1$ to plot the first two iterations (see appendix for the code):

```
library("plotrix")
plot(0.500000,  0.50000000, pch='0', xlim=c(-0.5,1.5 ), asp=1, ylim=c(-0.5,2), xlab="X", ylab=
"Y")
points(1.500000,0.00000000, pch='1')
points(1.083333,    -0.08333333, pch='2')
points(0,1, pch=19, col="orange", lwd=5)
points(1,0, pch=19, col="orange", lwd=5)
abline(1,-1, col="skyblue")
abline(h=0, col="red")
abline(v=0, col="red")
draw.circle(1,1, radius=1, border="palegreen1")
```



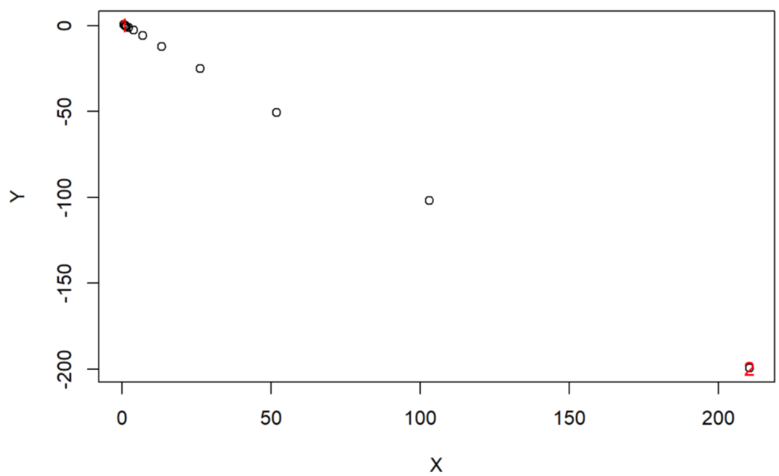
In the plot above, the orange points indicate the exact solutions to the system. Point 0 denotes our initial guess, while points 1 and 2 denote the solution approximations corresponding to first and second iterations when Newton's method is applied.

d. (Code) Write a code to solve this problem and plot the trajectory of solution in a xy plane, for $N = 30$ iterations.

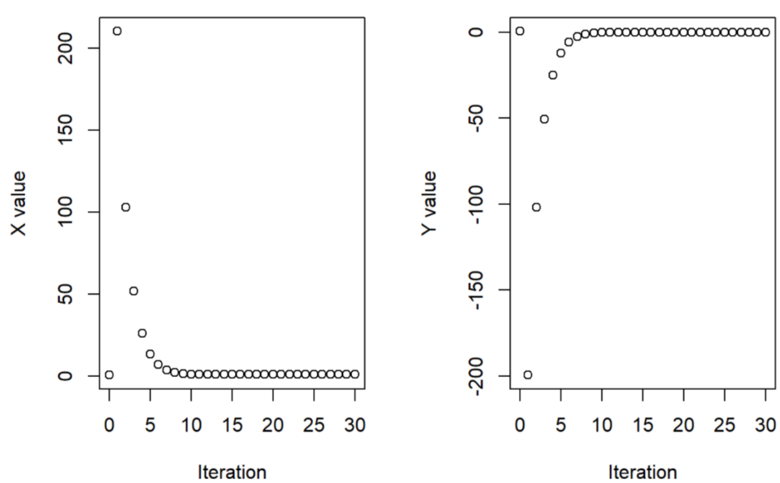
```
f1<-function(x,y){
  return((x-1)^2+(y-1)^2-1)
}
f2<-function(x,y){
  return((x+y-1))
}
J1<-function(x,y){
  return(2*(x-1))
}
J2<-function(x,y){
  return(2*(y-1))
}
Newton<-function(x,y, f1, f2, J1, J2,N, epsilon){
  X<-c(x)
  Y<-c(y)
  for(i in 1:N){
    Jac<-matrix(c(J1(X[i],Y[i]), J2(X[i], Y[i]),1,1), ncol=2, byrow = T)
    #Check if determinant is 0
    if(((Jac[1,1]*Jac[2,2])-(Jac[1,2]*Jac[2,1]))==0){
      Jac<-Jac+epsilon*matrix(c(1,0,0,1), ncol=2, byrow = T)
    }
    V<-matrix(c(X[i], Y[i]), ncol=1)-solve(Jac)%*%matrix(c(f1(X[i],Y[i]), f2(X[i],Y[i])))
    X[i+1]<- V[1]
    Y[i+1]<- V[2]
  }
  value<-data.frame(X,Y)
  return(value)
}
```

We know run the algorithm for two values of epsilon, 0.05 and 1. Moreover, to see that the approximations converge to either of the exact solutions, we separately plot the trajectory of each coordinate given the number of iterations. When $\epsilon = 0.05$, we see that after first iteration we move far away from the true solution (see point 2 in red), but eventually approach it. On the other hand, when $\epsilon = 1$, we have better approximations overall, since we observe little changes in x and y coordinates as we iterate further.

```
Values<-Newton(1/2,1/2, f1, f2, J1, J2, 30, 0.05)
plot(Values[,1], Values[,2], xlab = "X", ylab="Y")
points(Values[1,1],Values[1,2], pch='1', lwd=5, col="red")
points(Values[2,1],Values[2,2], pch='2', lwd=5, col="red")
```




```
par(mfrow=c(1,2))
plot(0:30, Values[,1], xlab = "Iteration", ylab="X value")
plot(0:30, Values[,2], xlab = "Iteration", ylab="Y value")
```




```
par(mfrow=c(1,1))
Values2<-Newton(1/2,1/2, f1, f2, J1, J2, 30, 1)
plot(Values2[,1], Values2[,2], xlab = "X", ylab="Y")
```