Optimization Methods I.
Newton-Raphson and others.
Exercises.

Anatoli Iambartsev
IME-USP

Newton-Raphson I-dim. Finding a root.

Consider the Newton method (sometimes called also tangent method) in one-dimensional case in order to solve the equation $g(\theta) = 0$. Suppose that g is differentiable. Let $\bar{\theta}$ be a root of the equation. By Taylor's expansion obtain

$$0 = g(\bar{\theta}) = g\big(\theta_n + (\bar{\theta} - \theta_n)\big) = g(\theta_n) + (\bar{\theta} - \theta_n) g'(\xi),$$

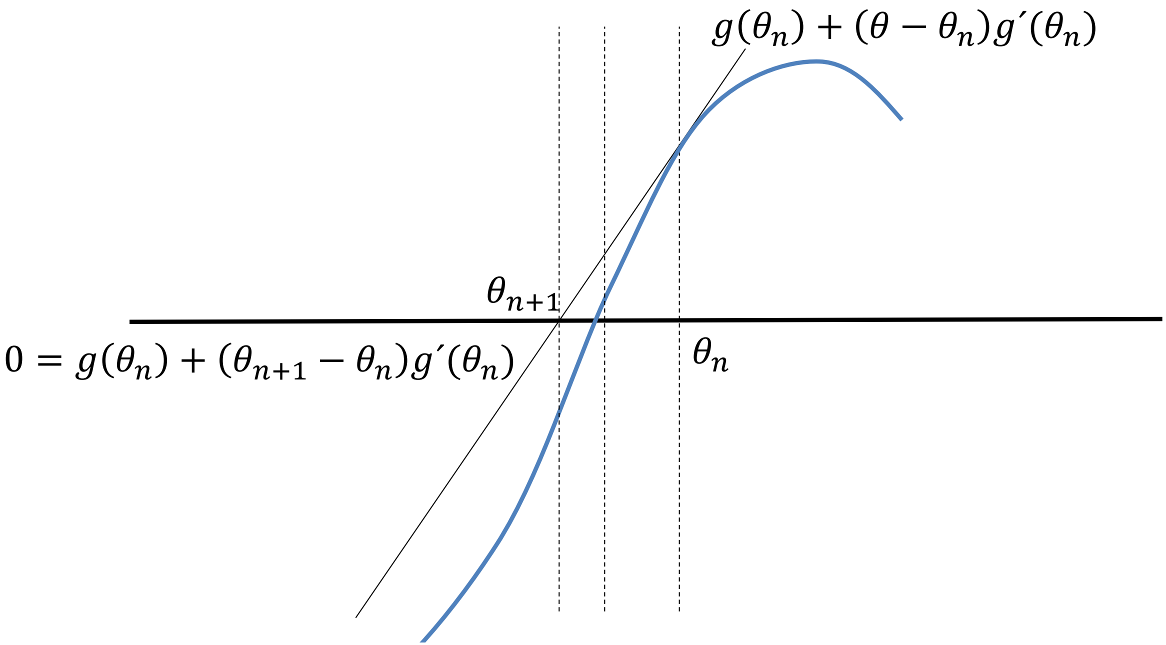
$$0 \cong g(\theta_n) + (\bar{\theta} - \theta_n) g'(\theta_n)$$

$$0 \cong \frac{g(\theta_n)}{g'(\theta_n)} + (\bar{\theta} - \theta_n) \Longrightarrow \bar{\theta} = \theta_n - \frac{g(\theta_n)}{g'(\theta_n)}$$

$$\Longrightarrow \theta_{n+1} = \theta_n - \frac{g(\theta_n)}{g'(\theta_n)}.$$

Newton-Raphson I-dim. Finding a root.

$$\theta_{n+1} = \theta_n - \frac{g(\theta_n)}{g'(\theta_n)}$$



Newton-Raphson I-dim. Finding a minimum.

In order to find the minimum of a function $f(\theta)$ the N-R method use Taylor's expansion of a second order:

$$f(\theta_n + p) \cong f(\theta_n) + pf'(\theta_n) + \frac{1}{2}p^2 f''(\theta_n) =: m(p) \to \min_p$$

$$\frac{\partial m(p)}{\partial p} = f'(\theta_n) + pf''(\theta_n) = 0.$$

Thus

$$p = -\frac{f'(\theta_n)}{f''(\theta_n)} \quad \text{and} \quad \theta_{n+1} = \theta_n - \frac{f'(\theta_n)}{f''(\theta_n)}.$$

Newton-Raphson I-dim. Finding a root and minimum.

$$g(\theta) = 0 \qquad \text{vz} \qquad f(\theta) \to \min$$

$$\theta_{n+1} = \theta_n - \frac{g(\theta_n)}{g'(\theta_n)} \qquad \theta_{n+1} = \theta_n - \frac{f'(\theta_n)}{f''(\theta_n)}$$

Optimization problems:

(i) To find an extreme points of a function $h(\theta)$ in a domain $\theta \in \Theta$.

(ii) To find a solution (solutions) of an equation $g(\theta) = 0$ in a domain $\theta \in \Theta$.

Two type of problem can be considered as equivalent:

$(i) \to (ii)$ Reformulate the problem (ii) in the form of (i) by choosing $h(\theta) = g^2(\theta)$.

$(ii) \to (i)$ Reformulate the problem (i) in the form of (ii) by choosing $g(\theta) = \frac{dh(\theta)}{d\theta}$.

Example. Newton-Raphson I-dim. Finding a root.

Consider the case $g(\theta) = \theta^2$: for some $a > 0$ consider equation $\theta^2 - a = 0$. Let us apply the recurrent formula

$$\theta_{n+1} = \theta_n - \frac{g(\theta_n)}{g'(\theta_n)}.$$

Thus

$$\theta_{n+1} = \theta_n - \frac{\theta_n^2 - a}{2\theta_n} = \frac{1}{2}\Big(\theta_n + \frac{a}{\theta_n}\Big).$$

(Heron's formula)

One says that a calculator uses the Heron's formula in order to compute a square root of a number.

Example. Newton-Raphson I-dim. Finding a square root.

The iteration formula

$$\theta_{n+1} = \frac{1}{2}\Big(\theta_n + \frac{a}{\theta_n}\Big).$$

does not used in calculators according https://www.quora.com/How-do-computers-compute-the-square-root-of-something

"On modern computer hardware, it is much cheaper to perform a multiplication operation than a division operation... So one typical trick is this: Instead of computing the square root, compute the reciprocal of the square root. That is, instead of $\sqrt{n}$, compute $1/\sqrt{n}$. It turns out that this is a far easier number to compute, and if you need the square root, multiply this number by n and you are done."

Example. Newton-Raphson I-dim. Finding a square root.

Consider equation $g(\theta) = 1/\theta^2$: for some $a > 0$ consider equation $1/\theta^2 - a = 0$. Let us apply the recurrent formula

$$\theta_{n+1} = \theta_n - \frac{g(\theta_n)}{g'(\theta_n)}.$$

Thus

$$\theta_{n+1} = \theta_n - \frac{\frac{1}{\theta_n^2} - a}{-\frac{2}{\theta_n^3}} = \frac{1}{2}\Big(3\theta_n - a\theta_n^3\Big).$$

Newton-Raphson I-dim. Convergence conditions.

Consider an iteration formula $\theta_{n+1} = \phi(\theta_n)$. Let again $\bar{\theta}$ be a root $g(\bar{\theta}) = 0$, and suppose that g is a differentiable function, thus

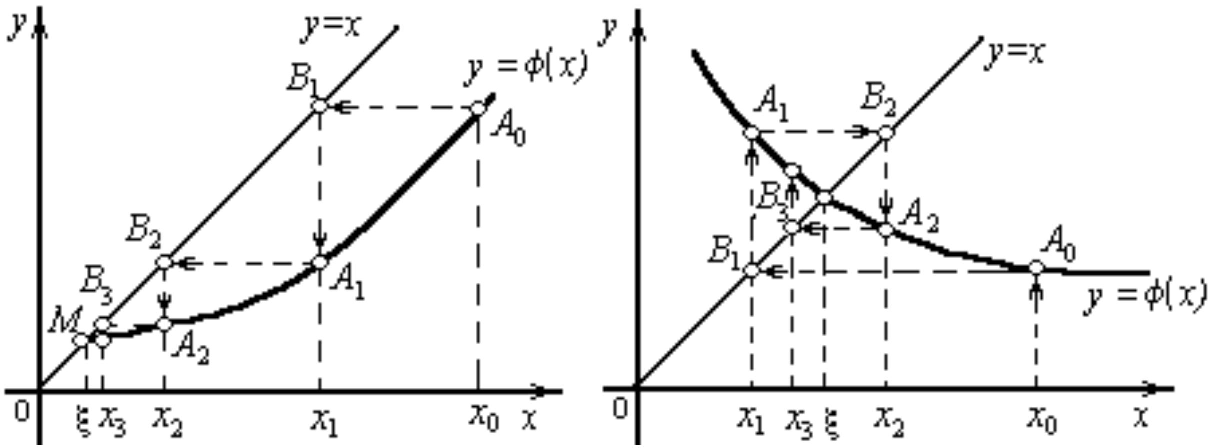
$$\theta_{n+1} - \bar{\theta} = \phi(\theta_n) - \phi(\bar{\theta}) = (\theta_n - \bar{\theta})\phi'(\xi),$$

where ξ is between $\bar{\theta}$ and θ_n.

If $|\phi'(\theta)| \leqslant q < 1$ for all $\theta \in \mathbb{R}$, then $|\theta_n - \bar{\theta}|$ decrease as at least geometrical sequence with progression coefficient $q < 1$. If $|\phi'(\bar{\theta})| > 1$, then $|\phi'(\theta)| > 1$ at some neighborhood of $\bar{\theta}$ and iterations will not converge. If $|\phi'(\bar{\theta})| < 1$ and $|\phi'(\theta)| > 1$ out of some neighborhood of $\bar{\theta}$, then we need to start near the point $\bar{\theta}$ in order to achieve a convergence.

Note: (i) lessen q faster convergence, (ii) near $\bar{\theta}$ the convergence is determined by $\phi'(\bar{\theta})$ and it is faster if $\phi'(\bar{\theta}) = 0$.

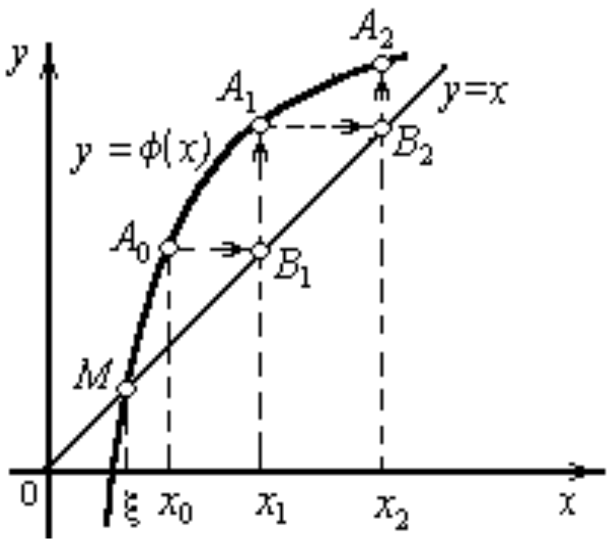
Successive approximation method. The first step of the method substitutes the equation $g(\theta) = 0$ by an equivalent version $\theta = \phi(\theta)$. The second step is the iteration scheme: (1) $\theta = \theta_0$; (2) then $\theta_{n+1} = \phi(\theta_n), n = 0, 1, \ldots$.



Note, $|\phi'(\theta)| < 1$ on the pictures.

Successive approximation method. Condition for convergence.

If $|\phi'(\theta)| > 1$ then the method can diverge, see on the following picture:



Successive approximation method. Criteria for convergence.

Let $\phi(\theta)$ defined and differentiable on the interval $[a,b]$ with values on $[a,b]$. If there exists $q > 0$, such that $|\phi'(\theta)| \leqslant q < 1$ for all $\theta \in [a,b]$, then

1. iteration process $\theta_{i+1} = \phi(\theta_i)$ converges independently of the initial choice $\theta_0 \in [a,b]$;

2. the limiting value $\xi = \lim_{n\to\infty} \theta_n$ is the unique root of the equation $\theta = \phi(\theta)$ on the interval $[a,b]$.

Successive approximation method. Example.

Consider the equation $g(x) = 0$, with $g(x) = x^3 - x - 1$. Note that the equation has a root on the interval $[1, 2]$, because $g(1) = -1 < 0$ and $g(2) = 5 > 0$. The equivalent equation is $x = x^3 - 1$, with $\phi(x) = x^3 - 1$. Observe that $\phi'(x) = 3x^2 > 3$ for any $x \in [1, 2]$. Thus the process will diverge.

Another equivalent equation is $x = \sqrt[3]{x+1}$, and $\phi'(x) = \frac{1}{3\sqrt[3]{(x+1)^2}} < \frac{1}{4}$ for any $x \in [1, 2]$. The process converges very fast.

[RC] Example 5.2. Newton-Raphson 2-dim in R: "nlm" function.

The likelihood associated with the mixture model

$$0.25N(\mu_1, 1) + 0.75N(\mu_2, 1)$$

is bimodal. For a simulated sample of 400 observations from this mixture with $\mu_1 = 0, \mu_2 = 2.5$, it is produced (minus likelihood) by

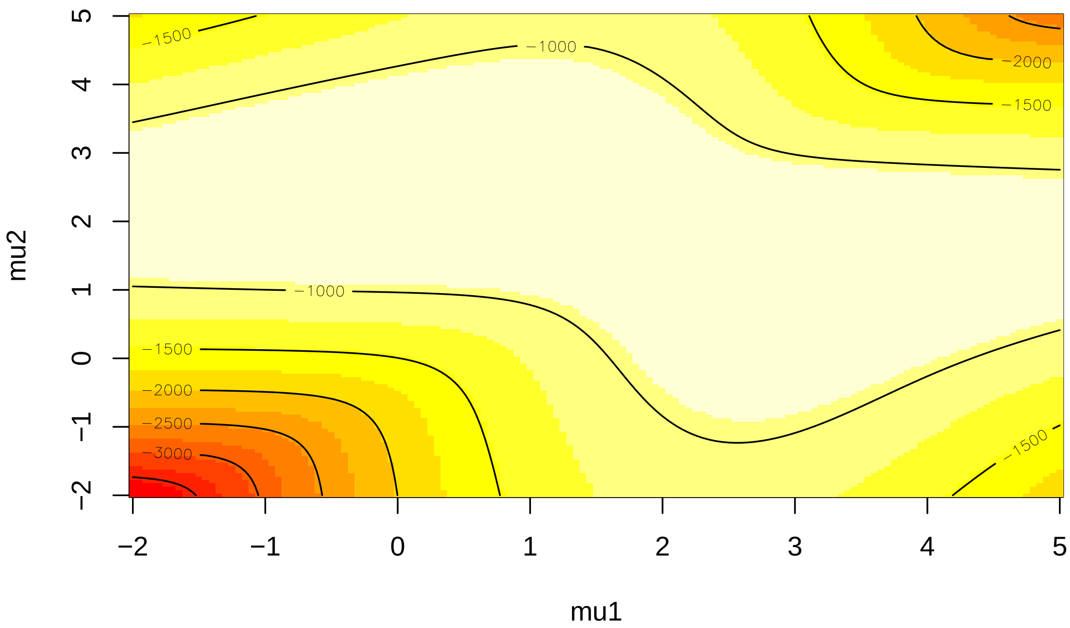
```
> set.seed(1)

> da=c(rnorm(100),2.5+rnorm(300))

> like=function(mu)-sum(log((.25*dnorm(da-mu[1])+
.75*dnorm(da-mu[2]))))
```

[RC] Example 5.2. Newton-Raphson 2-dim in R: "nlm" function.

By "contour" and "image" the log-likelihood function "like" is computed.

```
> mu1=seq(-2,5,by=0.05) #alpha grid for image

> mu2=seq(-2,5,by=0.05) #beta grid for image

> post=matrix(ncol=length(mu1),nrow=length(mu2))

> for (i in 1:length(mu1)){ for (j in 1:length(mu2))

>     { post[i,j]=-like(c(mu1[i],mu2[j])) } }

> image(mu1,mu2,post,xlab="mu1",ylab="mu2")

> contour(mu1,mu2,post,add=T)
```

[RC] Example 5.2. Newton-Raphson 2-dim in R: “nlm” function.

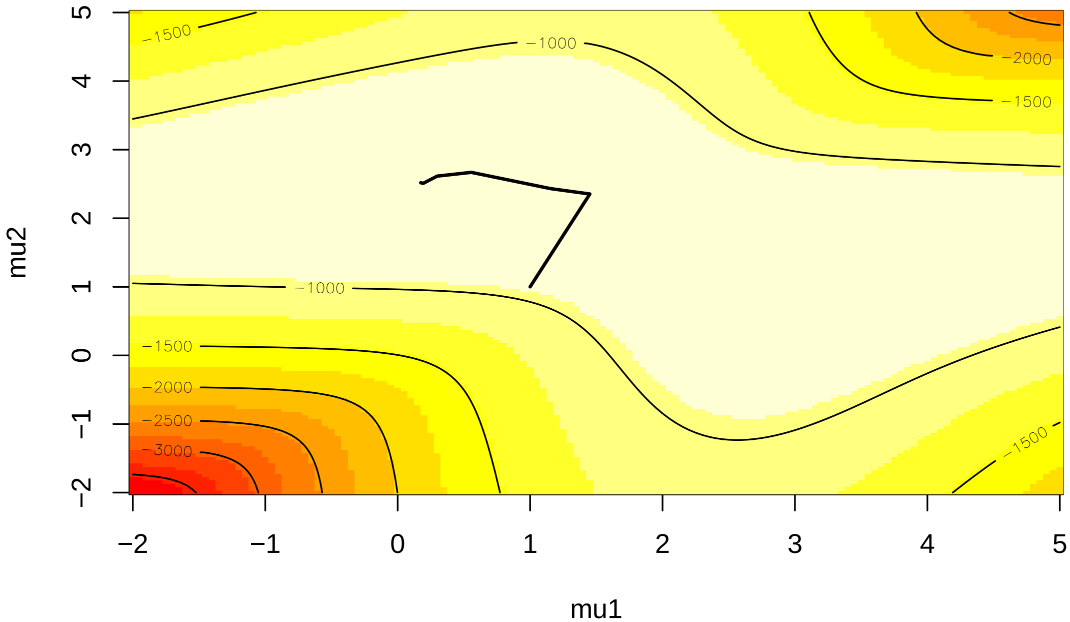


[RC] Example 5.2. Newton-Raphson 2-dim in R: “nlm” function.

Using “nlm” the models are obtained within a few iterations, depending on the starting points, and the intermediate values of the Newton-Raphson sequence can be plotted by

```
> sta=c(1,1)
> mmu=sta
> for (i in 1:(nlm(like,sta)$it)) {
>   mmu=rbind(mmu,nlm(like,sta,iterlim=i)$est)
>   }
> lines(mmu,pch=19,lwd=2)
```

[RC] Example 5.2. Newton-Raphson 2-dim in R: "nlm" function.



[RC] Example 5.2. Newton-Raphson 2-dim in R: "nlm" function.

```
> nlm(like,sta)
$minimum
[1] 695.8622
$estimate
[1] 0.1740339 2.5178156
$gradient
[1] 5.684342e-07 -1.219130e-06
$code
[1] 1
$iterations
[1] 8
Mensagens de aviso perdidas:
In nlm(like, sta) : NA/Inf substituido pelo máximo valor positivo
```

[RC] Example 5.2. Newton-Raphson 2-dim in R: "optim" function.

optim(par, fn, gr = NULL, ...,

method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", "Brent"),

lower = -Inf, upper = Inf,

control = list(), hessian = FALSE)

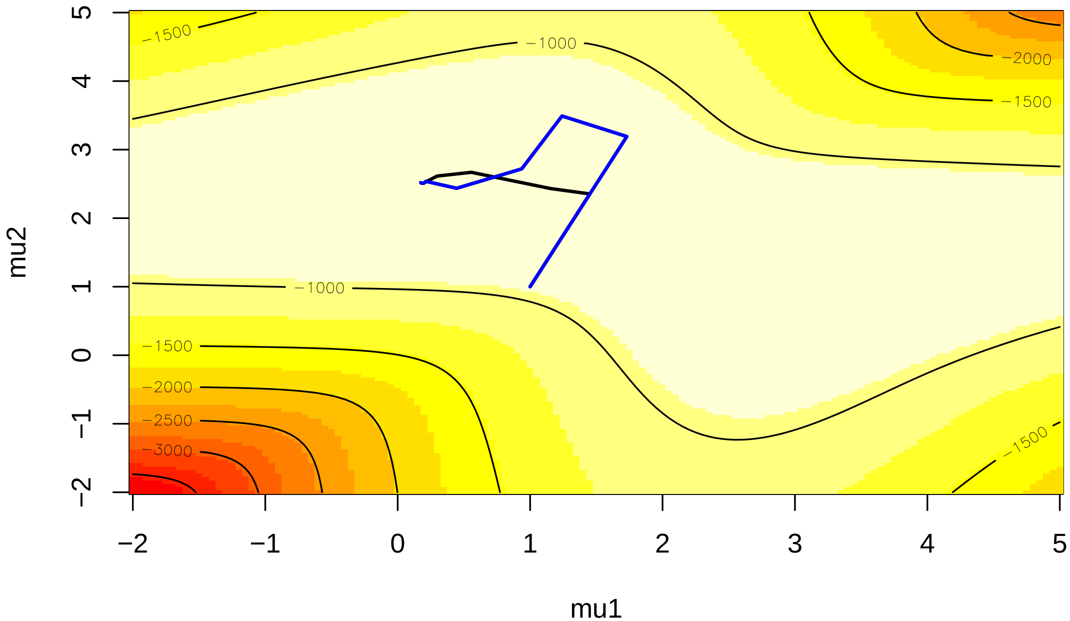
[RC] Example 5.2. Newton-Raphson 2-dim in R: "optim" function.

```
> optim(c(1,1), like, method = "BFGS")

$par

[1] 0.174035 2.517817

$value

[1] 695.8622

$counts

function gradient

31 9

$convergence

[1] 0

$message

NULL
```

[RC] Example 5.2. Newton-Raphson 2-dim in R: “optim” function.

```
n=as.numeric(optim(c(1,1), like, method ="BFGS")$count[1])

sta=c(1,1)

mmu=sta

for (i in 1:n) {

mmu=rbind(mmu,optim(c(1,1), like, method ="BFGS",
control=list(maxit=i))$par)}

lines(mmu,pch=19,lwd=2,col="blue")
```

[RC] Example 5.2. Newton-Raphson 2-dim in R: "optim" function.



Newton-Raphson for system of equations.

Consider the system of equations, for $\theta \in \mathbb{R}^d$

$$\Phi(\theta) = 0, \text{ where } \Phi : \ \mathbb{R}^d \to \mathbb{R}^d.$$

The method is derived from the equation

$$\Phi(\theta_n) + \Phi'(\theta_n)(\theta_{n+1} - \theta_n) = 0,$$

and

$$\theta_{n+1} = \theta_n - (\Phi'(\theta_n))^{-1}\Phi(\theta_n).$$

Newton-Raphson for system of equations. Bi-dimensional case.

Consider the system of equations, for $\theta = (\theta_1, \theta_2) \in \mathbb{R}^2$

$$\Phi(\theta) = \begin{pmatrix} \Phi_1(\theta) \\ \Phi_2(\theta) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \text{ where } \Phi: \ \mathbb{R}^2 \to \mathbb{R}^2,$$

the iteration formula is

$$\begin{pmatrix} \theta_1^{(n+1)} \\ \theta_2^{(n+1)} \end{pmatrix} = \begin{pmatrix} \theta_1^{(n)} \\ \theta_2^{(n)} \end{pmatrix} - \begin{pmatrix} \frac{\partial \Phi_1(\theta)}{\partial \theta_1} & \frac{\partial \Phi_1(\theta)}{\partial \theta_2} \\ \frac{\partial \Phi_2(\theta)}{\partial \theta_1} & \frac{\partial \Phi_2(\theta)}{\partial \theta_2} \end{pmatrix}^{-1}_{\theta=\theta^{(n)}} \begin{pmatrix} \Phi_1(\theta^{(n)}) \\ \Phi_2(\theta^{(n)}) \end{pmatrix}.$$

Newton-Raphson for system of equations. Bi-dimensional case. Example.

Find the minimum $f(x) = x_1^2 + 2x_1x_2 + 3x_2^2 + 4x_1$. Make the first step of Newton-Raphson method.

After differentiation we have

$$\begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 + 2x_2 + 4 \\ 2x_1 + 6x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Initial $x^(0) = (0, 0)^T$. Thus

$$\begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 & 2 \\ 2 & 6 \end{pmatrix}^{-1} \begin{pmatrix} 4 \\ 0 \end{pmatrix} = -\begin{pmatrix} \frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} 4 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \end{pmatrix}.$$