Contents

1 Solutions for Chapter 1 **2**

2 Solutions for Chapter 2 **3**

3 Solutions for Chapter 3 **5**

4 Solutions for Chapter 4 **7**

5 Solutions for Chapter 5 **10**

6 Solutions for Chapter 6 **13**

7 Solutions for Chapter 7 **18**

1 Solutions for Chapter 1

Prob. 1: Define the point-to-set mapping on $\mathbb{R}^n$ by

$$A(\mathbf{x}) = \{\mathbf{y} : \mathbf{y}'\mathbf{x} \le b\},$$

where b is a fixed constant. Is A closed?

Solution: Yes, it is. Let $\mathbf{x}_n$ be a sequence that converges to $\mathbf{x}$ and let $\mathbf{y}_n$ be a sequence that converges to $\mathbf{y}$ and satisfies that $\mathbf{y}_n'\mathbf{x}_n \le b$ for all n. Since the inner product is a continuous operator we obtain that $\mathbf{y}'\mathbf{x} \le b$ as required. To add more details, assume the contrary, i.e. the existence of $\epsilon > 0$ such that $\mathbf{y}'\mathbf{x} > b + \epsilon$. However,

$$|\mathbf{y}_n'\mathbf{x}_n - \mathbf{y}'\mathbf{x}| \le |\mathbf{y}_n'\mathbf{x}_n - \mathbf{y}'\mathbf{x}_n| + |\mathbf{y}\mathbf{x}_n - \mathbf{y}'\mathbf{x}| \le \|\mathbf{y}_n - \mathbf{y}\|\|\mathbf{x}_n\| + \|\mathbf{y}\|\|\mathbf{x}_n - \mathbf{x}\|$$

and the contradiction now follows from the assumptions on convergence and the fact that a converging sequence is bounded.

Prob. 2: Consider the iterative process

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right),$$

where $a > 0$. Assuming the process converges, to what does it converge? What is the order of convergence.

Solution: The solutions of $x = (1/2)(x + a/x)$ is $x^* = \pm a^{1/2}$. If we denote

$$f(x) = \frac{1}{2}\left(x + \frac{a}{x}\right),$$

then $f(x^*) = x^*$ and

$$f(x) \approx x^* + \frac{a}{4(x^*)^3}(x - x^*)^2.$$

It follows that the order of convergence is 2.

2 Solutions for Chapter 2

Prob. 1: Let $f(x) = ax^2 - 2bx + c$. Under which conditions does f has a minimum? What is the minimizing x?

Solution: Taking derivative we get $\dot{f}(x) = 2ax - 2b = 0$ as the first-order condition. The second derivative is $2a$. All points are inner points. Hence a nesesery and sufficient condition for $x^* = b/a$ to be the minimum is $a > 0$.

Prob. 2: Let $f(\mathbf{x}) = \mathbf{x}'\mathbf{A}\mathbf{x} - 2\mathbf{b}'\mathbf{x} + c$, with $\mathbf{A}$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector and c a scalar. Under which conditions does f has a minimum? a unique minimum? What is the minimizing $\mathbf{x}$?

Solution: Write

$$\mathbf{x}'\mathbf{A}\mathbf{x} - 2\mathbf{b}'\mathbf{x} + c = (\mathbf{x} - \mathbf{A}^{-1}\mathbf{b})'\mathbf{A}(\mathbf{x} - \mathbf{A}^{-1}\mathbf{b}) - \mathbf{b}'\mathbf{A}^{-1}\mathbf{b} + c$$

The point $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$ is a minimum if and only if $\mathbf{A}$ is positive-definite. If it is — it is obvious the point is a minimum. If it is not — one can find a direction along which one can approach $-\infty$.

Prob. 3: Write a function that finds the location and value of the minimum of a quadratic function.

Solution:

```
> minquad <- function(A,b,c)
+ {
+     x <- solve(A)%*%b
+     f <- t(x)%*%A%*%x -2*b%*%x + c
+     return(list(x=x,f=f)) + }
>
> A <- matrix(c(1,3,-1,2),2,2,byrow=TRUE)
> b <- c(5,2)
> c <- 1
> A
     [,1] [,2]
[1,]    1    3 [2,]   -1    2
> b
[1] 5 2
> c
[1] 1
```

```
> minquad(A,b,c)
$x
     [,1]
[1,]  0.8 [2,]  1.4

$f
     [,1]
[1,] -5.8
```

Prob. 4: Plot a contour plot of the function f with $\mathbf{A} = \begin{pmatrix} 1 & 3 \\ -1 & 2 \end{pmatrix}$, $\mathbf{b} = (5,2)'$ and $c = 1$. Mark, on the plot, the location of the minimum.

Solution:

```
> x <- y <- seq(-20,20,by=0.1);
> z <- outer(x,y,function(x,y)
+     x^2 + 2*x*y + 2*y^2 - 2*5*x - 2*2* y + 1)
> contour(x,y,z)
> points(8,-3,col=2)
```

3 Solutions for Chapter 3

Prob. 1: Find the minimum of the function `-humps`. Use different ranges.

Solution:

```
> humps <- function(x) 1/((x-0.3)^2+0.01)+1/((x-0.9)^2+0.04)-6
> x <- seq(-5,5,by=0.01)
> plot(x,humps(x),type="l")
> optimize(humps,c(-2,2),max=TRUE)
$maximum [1] 0.300377
$objective [1] 96.5014
> optimize(humps,c(0.31,2),max=TRUE)
$maximum [1] 0.8927036
$objective [1] 21.73457
> optimize(humps,c(0.5,1.5),max=TRUE)
$maximum [1] 0.8927257
$objective [1] 21.73457
> optimize(humps,c(0.2,1),max=TRUE)
$maximum [1] 0.3003759
$objective [1] 96.5014
```

Prob. 2:

1. Given $f(x_n), f'(x_n)$ and $f'(x_{n-1})$, show that

$$q(x) = f(x) + f'(x_n)(x - x_n) + \frac{f'(x_{n-1}) - f'(x_n)}{x_{n-1} - x_n} \cdot \frac{(x - x_n)^2}{2},$$

   has the same derivatives as f at x_n and x_{n-1} and is equal to f at x_n.

2. Construct a line search algorithm based on this quadratic fit.

Solution:

1. When $x = x_n$ the second and third terms in g vanish. Thus, $g(x_n) = f(x_n)$. Also

$$q'(x) = f'(x_n) + (f'(x_{n-1} - f'(x_n))\frac{x - x_n}{x_{n-1} - x_n}.$$

   plugging $x = x_n$ and $x = x_{n-1}$ gives the result.

2. The function g attains its minimum (if $[f'(x_{n-1})-f'(x_n)]/[x_{n-1}-x_n] > 0$.) at

$$x_{n+1} = x_n - \frac{f'(x_n)(x_{n-1} - x_n)}{f'(x_{n-1}) - f'(x_n)}.$$

Prob. 3: What conditions on the values and derivatives at two points guarantee that a cubic fit will have a minimum between the two points? Use the answer to develop a search scheme that is globally convergent for unimodal functions.

Solution: Assume $x_{n-1} < x_n$. A sufficient condition that a unimodal function has its minimum between these two points is $f'(x_{n-1}) < 0 < f'(x_n)$. Given a two-points pattern with this property, we can move to the next two-points pattern by selecting an interior point for which the value of the target function is strictly smaller and the pattern is sustained (with one of the two edge points. Using the function $Z(x_1, x_2) = f(x_1) + f(x_2)$ and the continuoity of the algorithm we get global convergence.

Prob. 4: Consider the function

$$f(x, y) = e^x(4x^2 + 2y^2 + 4xy + 2y + 1).$$

Use the function `fmin` to plot the function

$$g(y) = \min_x f(x, y).$$

Solution:

```
> fun2 <- function(x,y) exp(x)*(4*x^2+2*y^2+4*x*y+2*y+1)
> xy <- y <- x <- seq(-2,2,length=100)
> for (i in 1:length(y)) xy[i] <- optimize(fun2,c(-2,2),y=y[i])$min
> plot(y,fun2(xy,y),type="l")
> z <- outer(x,y,fun2)
> contour(x,y,z,nlev=200)
> lines(xy,y,col="red")
```

4 Solutions for Chapter 4

Prob. 1: Prove Theorem 4.2.2.

Solution: Consider a Taylor expansion of order two:

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)'\ddot{f}(x^*)(\mathbf{x} - \mathbf{x}^*) + o(\|\mathbf{x} - \mathbf{x}^*\|^2),$$

since $\dot{f}(x^*) = \mathbf{0}$. From the fact that

$$(\mathbf{x} - \mathbf{x}^*)'\ddot{f}(x^*)(\mathbf{x} - \mathbf{x}^*) \geq \lambda_{\min}(\ddot{f}(x^*))\|\mathbf{x} - \mathbf{x}^*\|^2,$$

where $\lambda_{\min}(\mathbf{A})$ is the smallest eigenvalue of the matrix $\mathbf{A}$ we obtain that there is a ball about $\mathbf{x}^*$ over which $f(\mathbf{x}) \geq f(\mathbf{x}^*) + \frac{1}{4}(\mathbf{x} - \mathbf{x}^*)'\ddot{f}(x^*)(\mathbf{x} - \mathbf{x}^*)$. Without loss of generality this ball is in the interior of Ω. The claim now follows from the fact that $(\mathbf{x} - \mathbf{x}^*)'\ddot{f}(x^*)(\mathbf{x} - \mathbf{x}^*) > 0$.

Prob. 2: To approximate the function g over the interval $[0, 1]$ by a polynomial h of degree n (or less), we use the criterion

$$f(\mathbf{a}) = \int_0^1 [g(x) - h(x)]^2 dx,$$

where $\mathbf{a} \in \mathbb{R}^{n+1}$ are the coefficients of h. Find the equations satisfied by the optimal solution.

Solution: Denote $h(x) = h_{\mathbf{a}}(x) = \mathbf{a}'\mathbf{x}$, with $\mathbf{a} = (a_0, \ldots, a_n)'$ and $\mathbf{x} = (1, x, x^2, \ldots, x^n)$. Thus,

$$\begin{aligned}(g(x) - h(x))^2 &= (g(x) - \mathbf{a}'\mathbf{x})(g(x) - \mathbf{a}'\mathbf{x})' \\ &= \mathbf{a}'\mathbf{x}\mathbf{x}'\mathbf{a} - 2g(x)\mathbf{x}'\mathbf{a} + g(x)^2.\end{aligned}$$

It follows that

$$\int_0^1 [g(x) - h(x)]^2 dx = \mathbf{a}'\mathbf{X}\mathbf{a} - 2\mathbf{b}'\mathbf{a} + c,$$

where $(\mathbf{X})_{ij} = \int_0^1 x^{i+j} dx = 1/(i + j + 1)$, $0 \leq i, j \leq n$, $(\mathbf{b})_i = \int_0^1 g(x)x^i dx$, and $c = \int_0^1 g(x)^2 dx$. Either from Question 3.8.2 or by taking derivatives, the first order equations are: $\mathbf{X}\mathbf{a} = \mathbf{b}$ or $\mathbf{a} = \mathbf{X}^{-1}\mathbf{b}$.

Prob. 3: Using first-order necessary conditions, find the minimum of the function

$$f(x,y,z) = 2x^2 + xy + y^2 + yz + z^2 - 6x - 7y - 8z + 9.$$

Verify the point is a relative minimum by checking the second-order conditions.

Solution: Taking derivatives we get

$$\dot{f}(x,y,z) = \begin{pmatrix} 4x + y - 6 \\ 2y + x + z - 7 \\ 2z + y - 8 \end{pmatrix} = 0.$$

The solution is $x^* = 6/5$, $y^* = 6/5$, $z^* = 17/10$. At the solution,

$$\ddot{f}(x^*,y^*,z^*) = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}.$$

The eigenvalues of this symmetric matrix are 4.4812, 2.6889, and 0.8299 — all positive. (Try with MATLAB!) The matrix is thus positive definite. The point is in the interior of Ω. From the second order sufficient conditions theorem it follows that the point is a local minimum.

Prob. 4: In control problem one is interested in finding numbers $u_0, \ldots, u_n$ that minimize the objective function

$$J = \sum_{k=0}^{n} \{(x_0 + u_0 + \cdots + u_{k-1})^2 + u_k^2\},$$

for a given x_0. Find the equations that determine the first order conditions.

Solution: Denote by $\mathbf{1}_k$ the vector of one at the kth coordinate and zeros elsewhere. Let $\mathbf{u} = (u_0, \ldots, u_n)'$. Thus,

$$\begin{aligned}
J &= x_0^2 + \sum_{k=1}^{n} \{(x_0 + \mathbf{u}'\mathbf{1}_{k-1})^2 + (\mathbf{u}'\mathbf{1}_k)^2\} \\
&= x_0^2 + \sum_{k=1}^{n} \{x_0^2 + 2x_0\mathbf{1}_{k-1}'\mathbf{u} + \mathbf{u}'\mathbf{1}_{k-1}\mathbf{1}_{k-1}'\mathbf{u} + \mathbf{u}'\mathbf{1}_k\mathbf{1}_k'\mathbf{u}\} \\
&= (n+1)x_0^2 + 2x_0\Bigg(\sum_{k=1}^{n} \mathbf{1}_{k-1}\Bigg)'\mathbf{u} + \mathbf{u}'\Bigg(\sum_{k=1}^{n} \{\mathbf{1}_{k-1}\mathbf{1}_{k-1}' + \mathbf{1}_k\mathbf{1}_k'\}\Bigg)\mathbf{u} \\
&= (n+1)x_0^2 + 2\mathbf{b}'\mathbf{u} + \mathbf{u}'\mathbf{A}\mathbf{u},
\end{aligned}$$

where

$$(\mathbf{A})_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j = 0, n \\ 2 & 1 \leq i = j < n, \end{cases}$$

$b_i = x_0$ when $i < n$, and $b_n = 0$. Finally, either from Question 3.8.2 or by taking derivatives, the first order equations are: $\mathbf{Au} = -\mathbf{b}$ or $\mathbf{u} = -\mathbf{A}^{-1}\mathbf{b}$.

5 Solutions for Chapter 5

Prob. 1: Investigate the function

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2.$$

Why doesn't the steepest decent algorithm converge?

Solution: We write the code of the function and its gradient. We then use the function `steepest.decent` that was presented in class in order to produce iterations of the algorithm:

```
> banana <- function(x) 100*(x[2]-x[1]^2)^2 + (1-x[1])^2
> banana.grad <- function(x)
+ {
+     fx1 <- -400*(x[2]-x[1]^2)*x[1] -2*(1-x[1])
+     fx2 <- 200*(x[2]-x[1]^2)
+     return(matrix(c(fx1,fx2),2,1))
+ }
> x0 <- xn <- x <- c(-1.9,2)
> fn <- banana(x)
> for (i in 1:30)
+ {
+     out <- steepest.decent(banana,banana.grad,x)
+     x <- out$x
+     xn <- cbind(xn,x)
+     fn <- c(fn,out$obj)
+ }
> fn
 [1] 267.62000000   0.61940091   0.51098122   0.51078300   6.43313982
 [6]   6.39978521   0.49922995   0.49910460   4.89353955   3.08695351
[11]   0.04347390   0.04143002   6.43184792   6.42924775   0.04321545
[16]   0.03682197   6.52399187   6.52154732   0.03705950   0.03697912
[21]   0.03689848   0.03681941   0.03674008   0.03666211   0.03658388
[26]   0.03650684   0.03642955   0.03635333   0.03627686   0.03620136
[31]   0.03612561
> xn
       xn
[1,] -1.9 1.705747 1.714492 1.714647 -1.535166 -1.527631 1.706145 1.706354
[2,]  2.0 2.944404 2.941679 2.940804  2.364528  2.323230 2.913358 2.912943
```

```
[1,] -1.210836 -0.7297018 1.208367 1.200266 -1.534420 -1.535257 1.207750
[2,]  1.473703  0.5016289 1.460906 1.437000  2.363699  2.361161 1.459404

[1,] 1.181528 -1.552509 -1.553361 1.192420 1.191730 1.192002 1.191320 1.191590
[2,] 1.389787  2.419606  2.417283 1.422449 1.421700 1.421449 1.420714 1.420464

[1,] 1.190915 1.191182 1.190513 1.190779 1.190115 1.190379 1.189719 1.189981
[2,] 1.419740 1.419492 1.418776 1.418529 1.417821 1.417574 1.416874 1.416628
```

Compare that to the actual minimum of the function:

```
> optim(x0,banana)
$par
[1] 0.9992542 0.9984668

$value
[1] 7.337513e-07

$counts
function gradient
     209       NA

$convergence
[1] 0

$message
NULL
```

Now let us plot the function, identify the initial point and the solution, and plot the path the algorithm went through:

```
> xx <- seq(-2,2,by=0.01)
> yy <- seq(-1,3,by=0.01)
> zz <- outer(xx,yy,function(x,y) 100*(y-x^2)^2 + (1-x)^2)
> contour(xx,yy,zz,nlev=30)
> points(x0[1],x0[2],col=2)
> text(x0[1],x0[2]+0.2,"start")
> points(1,1,col=2)
> text(1,1+0.2,"solution")
> lines(xn[1,],xn[2,],type="b",col=2)
```

Examining the iterations, we see that the algorithm moves from one side of the valley to the other side and then it moves very slowly towards the solution.

6 Solutions for Chapter 6

Prob. 1: Investigate the the performance of the DFP algorithm on the function

$$f(x,y) = 100(y-x^2)^2 + (1-x)^2$$

that was investigated in the previous homework.

Solution: We run, essentially, the same code that we used in the homework for the steepest decent. This time we use the function `DFP`, that is given in the notes:

```
> banana <- function(x) 100*(x[2]-x[1]^2)^2 + (1-x[1])^2
> banana.grad <- function(x)
+ {
+     fx1 <- -400*(x[2]-x[1]^2)*x[1] -2*(1-x[1])
+     fx2 <- 200*(x[2]-x[1]^2)
+     return(matrix(c(fx1,fx2),2,1))
+ }
> x0 <- xn <- x <- c(-1.9,2)
> S <- diag(2)
> fn <- banana(x)
> for (i in 1:20)
+ {
+     out <- DFP(banana,banana.grad,x,S)
+     x <- out$x
+     S <- out$S
+     xn <- cbind(xn,x)
+     fn <- c(fn,out$obj)
+ }
> fn
 [1] 2.676200e+02 6.194009e-01 5.050069e-01 4.880408e-01 2.979314e-01
 [6] 2.949670e-01 2.270763e-01 2.163086e-01 1.325023e-01 8.962444e-02
[11] 8.054625e-02 6.505483e-02 4.256467e-02 1.844240e-02 7.760900e-03
[16] 3.034009e-03 6.427977e-04 4.353188e-05 1.362797e-06 2.358034e-09
[21] 1.179113e-13
> xn
       xn
[1,] -1.9 1.705747 1.710581 1.698323 1.516109 1.513588 1.409638 1.406992
[2,]  2.0 2.944404 2.926989 2.886265 2.280821 2.273286 1.962733 1.957117
```

```
[1,] 1.358031 1.298304 1.265166 1.238358 1.161498 1.133352 1.076829 1.027723
[2,] 1.850818 1.683065 1.590529 1.524454 1.336240 1.287056 1.155249 1.051455

[1,] 1.025323 1.003134 1.001108 1.000018 1.000000
[2,] 1.051411 1.005697 1.002255 1.000031 1.000001
> optim(x0,banana)
$par
[1] 0.9992542 0.9984668
$value
[1] 7.337513e-07
$counts
function gradient
     209       NA
$convergence
[1] 0
$message
NULL
> xx <- seq(-2,2,by=0.01)
> yy <- seq(-1,3,by=0.01)
> zz <- outer(xx,yy,function(x,y) 100*(y-x^2)^2 + (1-x)^2)
> contour(xx,yy,zz,nlev=30)
> points(x0[1],x0[2],col=2)
> text(x0[1],x0[2]+0.2,"start")
> points(1,1,col=2)
> text(1,1+0.2,"solution")
> lines(xn[1,],xn[2,],type="b",col=2)
```

Observe that convergence was obtained after 20 steps, which all went in the right direction, as can be seen in the plot that is generated by the code.

Prob. 2: Investigate the rate of convergence of the algorithm

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [\delta I + (\ddot{f}(\mathbf{x}_n))^{-1}]\dot{f}(\mathbf{x}_n).$$

What is the rate if δ is larger than the smallest eigenvalue of $(\ddot{f}(\mathbf{x}^*))^{-1}$?

Solution: Repeating the arguments that were presented in the context of the steepest decent algorithm and repeated in the context of the quazi-Newtonian algorithms we get that if the function is quadratic then the rate is govern by the eigenvalues of the matrix SQ, where $S = \delta I + (\ddot{f}(\mathbf{x}_n))^{-1}$ and $Q = \ddot{f}(\mathbf{x}_n)$. Observe that in this case $SQ = I + \delta \ddot{f}(\mathbf{x}_n)$. The eigenvalues of QS are $\{1+\delta\lambda_i : i=1,\ldots,d\}$, where $\{\lambda_i\}$ are the eigenvalues of $\ddot{f}(\mathbf{x}_n)$. Based on Kantarovich' inequality one gets that the rate is bounded by

$$\Big(\frac{\delta(\lambda_{(1)}-\lambda_{(d)})}{2+\delta(\lambda_{(1)}+\lambda_{(d)})}\Big)^2 .$$

A general function may be approximated by a quadratic function in the vicinity of the solution. Hence, the above bound is an approximate bound once the algorithm approaches the solution.

Prob. 3: Use the formula

$$[A+\mathbf{b}\mathbf{a}']^{-1} = A^{-1} - \frac{A^{-1}\mathbf{a}\mathbf{b}'A^{-1}}{1+\mathbf{b}'A^{-1}\mathbf{a}},$$

in order to get a direct updating formula for the inverse of H_n in the BFGS method.

Solution: Recall that the approximation of the hessian is updated using the recursion:

$$H_{n+1} = H_n + \frac{(\Delta_n\dot{f})(\Delta_n\dot{f})'}{(\Delta_n\dot{f})'(\Delta_n\mathbf{x})} - \frac{H_n(\Delta_n\mathbf{x})(\Delta_n\mathbf{x})'H_n}{(\Delta_n\mathbf{x})'H_n(\Delta_n\mathbf{x})} .$$

Assume that the inverse of H_n, denoted S_n, was computed and stored in the previous iteration. Based on the relation that was given in the formula we get that

$$\tilde{S}_{n+1} = \Big[H_n + \frac{(\Delta_n\dot{f})(\Delta_n\dot{f})'}{(\Delta_n\dot{f})'(\Delta_n\mathbf{x})}\Big]^{-1} = S_n - \frac{S_n(\Delta_n\dot{f})(\Delta_n\dot{f})'S_n}{(\Delta_n\dot{f})'(\Delta_n\mathbf{x}) + (\Delta_n\dot{f})'S_n(\Delta_n\dot{f})} .$$

Denote the inverse of $\tilde{S}_{n+1}$ by $\tilde{H}_{n+1}$. A second application of the same formula produces:

$$\begin{aligned} S_{n+1} &= \Big[\tilde{H}_{n+1} - \frac{H_n(\Delta_n\mathbf{x})(\Delta_n\mathbf{x})'H_n}{(\Delta_n\mathbf{x})'H_n(\Delta_n\mathbf{x})}\Big]^{-1} \\ &= \tilde{S}_{n+1} + \frac{\tilde{S}_{n+1}H_n(\Delta_n\mathbf{x})(\Delta_n\mathbf{x})'H_n\tilde{S}_{n+1}}{(\Delta_n\mathbf{x})'H_n(\Delta_n\mathbf{x}) - (\Delta_n\mathbf{x})'H_n'\tilde{S}_{n+1}H_n(\Delta_n\mathbf{x})} , \end{aligned}$$

which involves only products of matrices.

Prob. 4: Read the help file on the function "`optim`". Investigate the effect of supplying the gradients with the parameter "`gr`" on the performance of the procedure.

Solution:

We try the function `optim` with and without the gradient function:

```
> ?optim
> optim(x0,banana)
$par
[1] 0.9992542 0.9984668
$value
[1] 7.337513e-07
$counts
function gradient
     209       NA
$convergence
[1] 0
$message
NULL
> optim(x0,banana,banana.grad)
$par
[1] 0.9992542 0.9984668
$value
[1] 7.337513e-07
$counts
function gradient
     209       NA
$convergence
[1] 0
$message
NULL
> optim(x0,banana,banana.grad,method="BFGS")
$par
[1] 1 1
$value
[1] 3.222467e-26
$counts
function gradient
      85       31
$convergence
[1] 0
$message
NULL
```

Observe that it is not enough to provide the gradient. One needs to specify a method that uses the gradient in order to have an effect. Also, observe that using the gradient resulted in less applications of both functions and in better convergence.

7 Solutions for Chapter 7

Prob. 1: Consider the constraints $x_1 \geq 0$, $x_2 \geq 0$ and $x_2 - (x_1 - 1)^2 \leq 0$. Show that $(1, 0)$ is feasible but not regular.

Solution: The gradients of the active constraints at the point $(1, 0)$ are $(0, 1)$ (second constraint) and $(-2(1-1), 1) = (0, 1)$ (third constraint). They are linearly dependent, hence the point is not regular.

Prob. 2: Find the rectangle of given perimeter that has greatest area by solving the first-order necessary conditions. Verify that the second-order sufficient conditions are satisfied.

Solution: Let P be the perimeter. The problem can be formulated as a constraint minimization problem:

$$\begin{array}{rl} \text{minimize} & -xy \\ \text{subject to} & x + y = P/2;\ -x \leq 0;\ -y \leq 0. \end{array}$$

The minimum is obtained when the inequality constraints ar not active. The Lagrangian is $l(x, y, \lambda) = -xy + \lambda(x+y-P/2)$, which leads to the first order necessary conditions:

$$\begin{aligned} -y + \lambda &= 0 &(1)\\ -x + \lambda &= 0 &(2)\\ x + y - P/2 &= 0, &(3) \end{aligned}$$

the solution of which is $x = y = P/4$; i.e. a square.

The subspace M is given by $\{(x, y) : x + y = 0\} = \{(x, -x) : x \in \mathbb{R}\}$. The partial hessian of the Lagrangian is the matrix

$$\ddot{l}_{\mathbf{x}}(\mathbf{x}^*) = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

Consequently, for every $\mathbf{x} \in M$, $\mathbf{x}'M\mathbf{x} = -2xy = 2x^2 > 0$. Hence, the second order sufficient condition of minimization is satisfied.

Prob. 3: Three types of items are to be stored. Item A costs one dollar, item B costs two dollars and item C costs 4 dollars. The demand for the three items are independent and uniformly distributed in the range $[0, 3, 000]$. How many of each type should be stored if the total budget is 4,000 dollars?

Solution: Let X_i be the demand for item i, $i = 1, 2, 3$, and let x_i be its stored amount. The unsold leftover is $(x_i - X_i)^+$, which has the expected value of

$$\mathbb{E}(x_i - X_i)^+ = \frac{1}{3,000} \int_0^{x_i} (x_i - y) dy = \frac{x_i^2}{6,000},$$

when $x_i \leq 3,000$ and is equal to $x_i - 3,000 + 1,500 = x_i - 1,500$ otherwise.

Let us look for solutions that satisfy $0 \leq x_i \leq 3,000$, for $1 \leq i \leq 3$. The problem is to minimize $x_1^2 + 2x_2^2 + 4x_2^3$, subject to the constraint $x_1 + x_2 + x_3 = 4,000$. The first order conditions are

$$\begin{aligned} 2x_1 + \lambda &= 0 \\ 4x_2 + \lambda &= 0 \\ 8x_3 + \lambda &= 0 \\ x_1 + x_2 + x_3 &= 4,000. \end{aligned}$$

Therefore, $\lambda/2 + \lambda/4 + \lambda/8 = -4,000$ and thus $\lambda = -(8/7)\,4,000$. It follows that $x_1 = 2,285.71$, $x_2 = 1,142.86$ and $x_3 = 571.42$.

Prob. 4: Let A be an $n \times m$ matrix of rank m and let L be an $n \times n$ matrix that is symmetric and positive-definite on the subspace $M = \{\mathbf{y} : A\mathbf{y} = \mathbf{0}\}$. Show that the $(n + m) \times (n + m)$ matrix

$$\begin{bmatrix} L & A' \\ A & \mathbf{0} \end{bmatrix}$$

is non-singular.

Solution: Let $\mathbf{x} \in R^n$ and $\mathbf{y} \in R^m$ be two vectors. We need to show that the equation

$$\begin{bmatrix} L & A' \\ A & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} L\mathbf{x} + A'\mathbf{y} \\ A\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

implies $\mathbf{x} = \mathbf{0}$ and $\mathbf{y} = \mathbf{0}$. Multiplying by $\mathbf{x}$ from the left yeilds

$$\mathbf{x}'L\mathbf{x} + \mathbf{x}'A'\mathbf{y} = \mathbf{x}'L\mathbf{x} = \mathbf{0}, \tag{4}$$

since $\mathbf{x}'A' = (A\mathbf{x})' = \mathbf{0}'$. The matrix L is positive definite over M. The vector $\mathbf{x}$ belongs to M. Thus the second equation in (4) implies $\mathbf{x} = \mathbf{0}$. Consequently, $A'\mathbf{y} = \mathbf{0}$. Thus, $\mathbf{y} = \mathbf{0}$ since the rank of A is m.

Prob. 5: Maximize $14x - x^2 + 6y - y^2 + 7$ subject to $x + y \leq 2$, $x + 2y \leq 3$.

Solution: Start with the case where $J = \emptyset$ then

$$\begin{aligned} 2x - 14 &= 0 \\ 2y - 6 &= 0, \end{aligned}$$

which has the solution $x = 7$, $y = 3$. However, this point is not feasible. (For example, it does not satisfy the first constraint.

Consider, next, the case $J = \{1\}$. The first order conditions are

$$\begin{aligned} 2x - 14 + \lambda_1 &= 0 \\ 2y - 6 + \lambda_1 &= 0 \\ x + y &= 2, \end{aligned}$$

This produces $\lambda_1 = 8$, $y = -1$, and $x = 3$.

For $J = \{2\}$ the first order conditions are

$$\begin{aligned} 2x - 14 + \lambda_2 &= 0 \\ 2y - 6 + 2\lambda_2 &= 0 \\ x + 2y &= 3, \end{aligned}$$

Here we get that $\lambda_2 = 4$, $x = 5$, and $y = -1$, which is not feasible.

Finally, for $J = \{1, 2\}$ we get that $x = y = 1$ and

$$\begin{aligned} \lambda_1 + \lambda_2 &= 12 \\ \lambda_1 + 2\lambda_2 &= 4. \end{aligned}$$

As a result, $\lambda_2 = -8 < 0$ and the K-T conditions are not satisfied.

To summarize, only the case $J = \{1\}$ is consistent with the K-T conditions. As for the sufficient condition observe that

$$\ddot{l}_{\mathbf{x}}(\mathbf{x}^*) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix},$$

which is positive definite.