# Nonlinear Optimization Fall 2021
## HW 1 Sample Solutions

**Q1** (a) Observe the first partial derivative of $f$ at $x$ is

$$\frac{\partial f}{\partial x_i}(x) = \frac{\partial}{\partial x_i}\left[\frac{1}{2}\sum_{k,l} x_k H_{kl} x_l\right]$$

$$= \frac{1}{2}\left[\sum_{l} H_{jl} x_l + \sum_{k} x_k H_{kj}\right].$$

$$\Rightarrow \nabla f(x) = \frac{1}{2}\left(\begin{bmatrix}\Sigma_l H_{1l} x_l \\ \vdots \\ \Sigma_l H_{nl} x_l\end{bmatrix} + \begin{bmatrix}\Sigma_k x_k H_{k1} \\ \vdots \\ \Sigma_k x_k H_{kn}\end{bmatrix}\right)$$

$$\boxed{= \frac{1}{2}\left(Hx + H^T x\right).}$$

Further, the second partial derivatives are given by

$$\frac{\partial f}{\partial x_i \partial x_j}(x) = \frac{\partial}{\partial x_i \partial x_j}\left[\frac{1}{2}\sum_{k,l} x_k H_{kl} x_l\right]$$

$$= \frac{1}{2}(H_{ij} + H_{ji}).$$

$$\Rightarrow \boxed{\nabla^2 f(x) = \frac{1}{2}(H + H^T).}$$

For symmetric $H$, $H = H^T$ and so $\boxed{\nabla f(x) = Hx}$ and $\boxed{\nabla^2 f(x) = H.}$

(b) Using linearity of gradients and part (a), we have

$$\nabla f(x) = \nabla\left(b^T A x - \tfrac{1}{2} x^T A^T A x\right)$$
$$= \nabla(b^T A x) - \nabla\left(\tfrac{1}{2} x^T A^T A x\right)$$
$$\boxed{\begin{aligned} &= b^T A - A^T A x \\ &= A^T(b - Ax) \end{aligned}}$$

Optional Simplification

Moreover,
$$\nabla^2 f(x) = \nabla^2\left(b^T A x - \tfrac{1}{2} x^T A^T A x\right)$$
$$= \nabla^2(b^T A x) - \nabla^2\left(\tfrac{1}{2} x^T A^T A x\right)$$
$$\boxed{= 0 - A^T A}$$

(c) Recall the Chain Rule for $g: \mathbb{R} \to \mathbb{R}$ and $h: \mathbb{R}^n \to \mathbb{R}$

ensures
$$\nabla(g \circ h)(x) = g'(h(x))\, \nabla h(x),$$
and
$$\nabla^2(g \circ h)(x) = g''(h(x))\, \nabla h(x)\, \nabla h(x)^T + g'(h(x))\, \nabla^2 h(x).$$

Observe that $g(t) = \sqrt{t}$ and $h(t) = x^T I x$ gives
$$f(x) = (g \circ h)(x).$$

Hence and

$$\nabla f(x) = \frac{1}{2\|x\|_2} 2x = \frac{x}{\|x\|_2}$$
$$\nabla^2 f(x) = \frac{-1}{4\|x\|_2^3} \cdot 4xx^T + \frac{1}{2\|x\|_2} 2I = -\frac{xx^T}{\|x\|_2^3} + \frac{I}{\|x\|_2}.$$

when $x \neq 0$.

Importantly, $f$ is not differentiable at $0$.

(d) Again we can use the chain rule, now with

$$g(t) = \sqrt{t}$$
$$h(x) = \|Ax-b\|_2^2 = \|b\|_2^2 - 2b^TAx + x^TAAx.$$

From part (b), we know the gradient and Hessian of $h$ are

$$\nabla h(x) = 2A^T(Ax-b),$$
$$\nabla^2 h(x) = 2A^TA.$$

Hence,

$$\boxed{\begin{aligned} \nabla f(x) &= \frac{1}{2\|Ax-b\|_2} \cdot 2A^T(Ax-b) = \frac{A^T(Ax-b)}{\|Ax-b\|_2}, \\ \nabla^2 f(x) &= \frac{-1}{4\|Ax-b\|_2^3} \cdot 4\left(\left((Ax-b)^TA\right)^T\left(A^T(Ax-b)\right)\right)^T \\ &\quad + \frac{1}{2\|Ax-b\|_2} \cdot 2A^TA \\ &= \frac{-\left(\left((Ax-b)^TA\right)^T\left(A^T(Ax-b)\right)\right)^T}{\|Ax-b\|_2^3} + \frac{A^TA}{\|Ax-b\|_2}. \end{aligned}}$$

when $Ax \neq b$.

Importantly, $f$ is not differentiable when $Ax = b$.

$\boxed{\text{Q2}}$ (a) Observe that $f(x+s) - f(x) - \nabla f(x)^T s$

$$= \theta(1) - \theta(0) - \theta'(0)$$

$$= \int_0^1 \big(\theta'(t) - \theta'(0)\big)\,dt.$$

First, we upper bound this by noting

$$\begin{aligned}\theta'(t) - \theta'(0) &= (\nabla f(x+ts) - \nabla f(x))^T s \\ &\leq \|\nabla f(x+ts) - \nabla f(x)\|_2 \|s\|_2 \\ &\leq L t \|s\|_2^2.\end{aligned}$$

$$\Rightarrow \int_0^1 (\theta'(t) - \theta'(0))\,dt \leq L\|s\|_2^2 \int_0^1 t\,dt = \underline{\tfrac{1}{2} L \|s\|_2^2}.$$

Likewise, we can lower bound this as

$$\begin{aligned}\theta'(t) - \theta'(0) &\geq -\|\nabla f(x+ts) - \nabla f(x)\|_2 \|s\|_2 \\ &\geq -L t \|s\|_2^2\end{aligned}$$

and so $\int_0^1 (\theta'(t) - \theta'(0))\,dt \geq -L\|s\|_2^2 \int_0^1 t\,dt = \underline{-\frac{L}{2}\|s\|_2^2}$. $\square$

(b) Applying the Fundamental Theorem of Calculus to $\theta'(t)$ implies $\theta'(t) = \theta'(0) + \int_0^t \theta''(\alpha)\,d\alpha$.

Plugging this in to the Fundamental Theorem for $\theta(t)$ gives

$$\underline{\theta(1) = \theta(0) + \theta'(0) + \int_0^1 \int_0^t \theta''(\alpha)\,d\alpha\,dt.}$$

Similar to part (a), observe that

$$f(x+s)-f(x)-\nabla f(x)^T s-\frac{1}{2}s^T\nabla^2 f(x)s$$
$$=\theta(1)-\theta(0)-\theta'(0)-\frac{1}{2}\theta''(0)$$
$$=\int_0^1\int_0^t\left(\theta''(\alpha)-\theta''(0)\right)d\alpha\, dt.$$

$\leftarrow$ Note the $\frac{1}{2}$ disappeared since $\int_0^1\int_0^t d\alpha\, dt=\frac{1}{2}$.

Then we can upper bound this by noting

$$\theta''(\alpha)-\theta''(0)=s^T\left(\nabla^2 f(x+\alpha s)-\nabla^2 f(x)\right)s$$
$$\leq \|\nabla^2 f(x+\alpha s)-\nabla^2 f(x)\|\,\|s\|_2^2$$
$$\leq Q\alpha\|s\|_2\|s\|_2^2=Q\alpha\|s\|_2^3.$$

$$\Rightarrow \int_0^1\int_0^t\left(\theta''(\alpha)-\theta''(0)\right)d\alpha\, dt\leq Q\|s\|_2^3\int_0^1\int_0^t\alpha\, d\alpha\, dt$$
$$=\frac{1}{6}Q\|s\|_2^3.$$

Likewise, we have the lowerbound as

$$\theta''(\alpha)-\theta''(0)\geq -\|\nabla^2 f(x+\alpha s)-\nabla^2 f(x)\|\,\|s\|_2^2$$
$$\geq -Q\alpha\|s\|_2^3.$$

$$\Rightarrow \int_0^1\int_0^t\left(\theta''(\alpha)-\theta''(0)\right)d\alpha\, dt\geq -Q\|s\|_2^3\int_0^1\int_0^t\alpha\, d\alpha\, dt$$
$$=\frac{-Q}{6}\|s\|_2^3.$$

$\square$

$\boxed{Q3}$ By the second order sufficient condition, we know $x^*$ is a strict local minimizer.

$\Rightarrow$ For some $\varepsilon > 0$, all $x \in B(x^*, \varepsilon) \setminus \{x^*\}$ have $f(x^*) < f(x)$ (strictly).

Suppose for contradiction that another global minimizer $y^*$ exists. Then $f(y^*) = f(x^*)$.

By convexity, every $\lambda \in [0,1]$ has

$$f(x^*) = \lambda f(x^*) + (1-\lambda) f(y^*) \geq f(\lambda x^* + (1-\lambda) y^*).$$

However, $f(x^*)$ is the minimum value of $f$,

$\Rightarrow$ This inequality must be equality and consequently every $\lambda x^* + (1-\lambda) y^*$ is also a global minimizer.

As $\lambda \to 1$, this gives global minimizers approaching $x^*$, contradicting it being strict.

Thus $x^*$ must be the unique global minimizer.

$\boxed{\text{Q4}}$ (a) Consider any pair of rubrics

$$x = (H_x, M_x, F_x)$$
$$y = (H_y, M_y, F_y)$$

and $\lambda \in [0,1]$.

Since $x$ is feasible,
$$\begin{cases} \lambda(H_x + M_x + F_x) \leq \lambda 100 \\ \lambda H_x, \lambda M_x \geq \lambda 15 \\ \lambda F \geq \lambda M_x \\ 50\lambda \leq \lambda(M_x + F_x) \leq 80\lambda \\ \lambda(H_x + M_x + F_x) \geq 90\lambda \end{cases}$$

Likewise $y$ is feasible,
$$\begin{cases} (1-\lambda)(H_y + M_y + F_y) \leq (1-\lambda)100 \\ (1-\lambda)H_y, (1-\lambda)M_y \geq (1-\lambda)15 \\ \vdots \\ (1-\lambda)(H_y + M_y + F_y) \geq (1-\lambda)90 \end{cases}$$

Adding up each pair of rescaled constraints shows $z = \lambda x + (1-\lambda)y = (H_z, M_z, F_z)$ is feasible,
$$\begin{cases} \lambda H_x + (1-\lambda)H_y + \lambda M_x + (1-\lambda)M_y + \lambda F_x + (1-\lambda)F_y \leq 100 \\ \lambda H_x + (1-\lambda)H_y, \ \lambda M_x + (1-\lambda)M_y \geq 15 \\ \vdots \\ \lambda H_x + (1-\lambda)H_y + \lambda M_x + (1-\lambda)M_y + \lambda F_x + (1-\lambda)F_y \geq 90. \end{cases}$$

Thus $\mathcal{P}$ is convex.

$\square$

(b) Note that $P$ is compact (closed since it is the intersection of seven closed halfspaces

bounded since $H \in [0,100]$, $M \in [0,100]$, $F \in [0,100]$).

Hence there must exist some $x^* = (H^*, M^*, F^*) \in P$

attaining $\sup_{x \in P} \{ C_H H + C_M M + C_F F + C_P(100 - H - M - F) \}$.

$$\sup_{x \in P} \{ C_H H + C_M M + C_F F + C_P(100 - H - M - F) \} = 100C_P + (C_H - C_P)H + (C_M - C_P)M + (C_F - C_P)F$$

Applying the given theorem at $x^*$ gives weights $\lambda_p \geq 0$ s.t.

$$\sum \lambda_p p = x^*, \quad \sum \lambda_p = 1.$$

$$\begin{aligned} \Rightarrow\ & 100C_P + (C_H - C_P)H^* + (C_M - C_P)M^* + (C_F - C_P)F^* \\ &= 100C_P + (C_H - C_P)\sum \lambda_p p_H + (C_M - C_P)\sum \lambda_p p_M + (C_F - C_P)\sum \lambda_p p_F \\ &= \sum \lambda_p \left(100C_P + (C_H - C_P)p_H + (C_M - C_P)p_M + (C_F - C_P)p_F\right). \end{aligned}$$

Thus the maximum obj value is a weighted average of the objective values at the corners $S$.

An average cannot be greater than all its elements

$\Rightarrow$ Some corner is at least as good as $x^*$.

Since $x^*$ is optimal, this corner must also be optimal. $\square$

In [14]:

```python
import numpy as np

#List of the corner points of the feasible region of all grading rubrics
#  (Adding in participation weights in as a fourth component)
S = np.array([[15,40,40,5],
              [20,40,40,0],
              [50,25,25,0],
              [40,25,25,10],
              [15,37.5,37.5,10],
              [15,15,65,5],
              [20,15,65,0],
              [50,15,35,0],
              [40,15,35,10],
              [15,15,60,10]])

#Function given a vector of scores (CH,CM,CF,CP) outputs optimal corner and score
def grade(C):
    corner_scores=np.matmul(S,C)/100 #Vector of scores at each corner
    p = np.argmax(corner_scores)     #Pick one of the corners that maximizes
    return (S[p], corner_scores[p])  #Return that corner and corresponding score

#Grading each of the students using this function
print( "Student 1's optimal corner and score...", grade(np.array([100,90,80,70])) )
print( "Student 2's optimal corner and score...", grade(np.array([85,85,85,85])) )
      #Note every corner is optimal for the second student
print( "Student 3's optimal corner and score...", grade(np.array([70,80,90,100])) )
```

```
Student 1's optimal corner and score... (array([50., 25., 25.,  0.]), 92.5)
Student 2's optimal corner and score... (array([15., 40., 40.,  5.]), 85.0)
Student 3's optimal corner and score... (array([15., 15., 60., 10.]), 86.5)
```