Prof. Dr. Karsten Urban
M. Sc. Lewin Ernst

Winter term 2024/2025
Sheet 14, 26 Points

Numerical Optimization
Solution to exercise sheet

review on 05.02.2024 during the exercise class

1. *(Projected Subgradient Method)*
For nonsmooth optimization we are forced to generalize the concept of classical derivatives of a function f at a point x_0. For this, we have introduced for convex functions f defined on a convex set the convex subdifferential ∂f, compare Definition 5.3.3.

   a) Calculate the following sub-differentials.

      i) $\partial f(0)$, for $f : \mathbb{R} \to \mathbb{R}$, $\qquad f(x) := \begin{cases} x^2, & x < 0, \\ x, & x \geq 0. \end{cases}$

      ii) $\partial f(0)$, for $f : \mathbb{R}^n \to \mathbb{R}$, $\quad f(x) := \|x\|_2 = \sqrt{x^T x}$.

   b) The Projected Subgradient Method is a method for non-smooth optimization, where (for example) the cost function f is not differentiable in the classical sense. In contrast to the general gradient method, the Projected Subgradient Method uses on the one hand an element of the convex sub-differential instead of it's classical gradient and on the other hand a projection of $x^{(k+1)}$ onto the feasible domain $\mathcal{F}$ (compare Algorithm 5.3.1). However, the choice of a reasonable step-size is not that trivial due to the missing smoothness of f.

      Consider now the unconstrained optimization problem

$$\min_{x \in \mathbb{R}} f(x), \tag{1}$$

      for $f(x) = |x|$ with its solution $x^* = 0$. Use the Projected Subgradient Method with $P_{\mathcal{F}} = id$ (because of $\mathcal{F} = \mathbb{R}^n$) for solving the optimization problem (1).

      Choose the step sizes

      i) $\sigma^{(k)} = \frac{1}{k+1}$, as well as

      ii) $\sigma^{(k)} = \frac{f(x^{(k)}) - f(x^*)}{|s^{(k)}|}$,

      and the initial vector $x^{(0)} = 2$ for computing $x^{(1)}, \ldots, x^{(5)}$ (until $s^{(k)} = 0$).

      Hint: You can use without a proof that the convex sub-differential of f in $x_0 = 0$ is given by $\partial f(0) = [-1, 1]$.

   c) Consider the optimization problem (1) with $f(x) = (|x| + 1)^2$ and its solution $x^* = 0$. Prove that for the Projected Subgradient Method with step size $\sigma^{(k)} = (f(x^{(k)}) - f(x^*))/|s^{(k)}|$, it holds

$$|x^{(k+1)} - x^*| \leq \frac{1}{2}|x^{(k)} - x^*|, \qquad k = 1, 2, \ldots.$$

(4 + 4 + 6 = 14 Points)

Solution:

a) i) With $x_0 = 0, f(x_0) = 0$, the subgradient condition becomes

$$\begin{aligned} f(y) \geq f(x_0) + s^\top (y - x_0), \qquad & \forall y \in \mathcal{F} = \mathbb{R}, \\ \iff f(y) \geq sy, \qquad & \forall y \in \mathbb{R}. \end{aligned}$$

We split the condition in

$$\begin{aligned} y^2 \geq sy, \qquad & \forall y \in \{y < 0\}, \\ \iff y \leq s, \qquad & \forall y \in \{y < 0\}, \end{aligned}$$

which gives $s \geq 0$. On the other hand we have

$$\begin{aligned} y \geq sy, \qquad & \forall y \in \{y \geq 0\} \\ \iff 1 \geq s, \qquad & \forall y \in \{y \geq 0\}, \end{aligned}$$

which gives $s \leq 1$. Together, we have

$$\partial f(0) = [0, 1].$$

ii) With $x_0 = 0, f(x_0) = 0$, the subgradient condition becomes

$$\begin{aligned} f(y) \geq f(x_0) + s^\top (y - x_0), \qquad & \forall y \in \mathcal{F} = \mathbb{R}^n, \\ \iff \sqrt{y^\top y} \geq s^\top y, \qquad & \forall y \in \mathbb{R}^n. \end{aligned}$$

This condition must also hold for the special case of $y = s$:

$$\begin{aligned} ||s||_2 = \sqrt{s^\top s} \geq s^\top s = ||s||_2^2, \\ \iff 1 \geq ||s||_2. \end{aligned}$$

And indeed, with $1 \geq ||s||_2$ and with Cauchy-Schwarz, we have

$$\sqrt{y^\top y} = ||y||_2 = \frac{||y||_2 ||s||_2}{||s||_2} \geq \frac{s^\top y}{||s||_2} \geq s^\top y, \quad \forall y \in \mathbb{R}^n.$$

Therefore it follows

$$\partial f(0) = \{s \in \mathbb{R}^n : ||s||_2 \leq 1\}.$$

b) i) The sub-differential of f is given by

$$\partial f(x) = \begin{cases} \{-1\}, & x < 0, \\ [-1, 1], & x = 0, \\ \{1\}, & x > 0, \end{cases} \tag{2}$$

and therefore for descent directions $d^{(k)} = \frac{-s^{(k)}}{||s^{(k)}||_2}$, for $x \neq 0$, it is

$$d^{(k)} = \begin{cases} 1, & x < 0, \\ -1, & x > 0. \end{cases} \tag{3}$$

With these observations it follows for the Projected Subgradient Method for $k = 0, 1 ... 5$ and $x^{(0)} = 2$:
$\rightarrow$ $s^{(0)} = -1$, $d^{(0)} = -1$, $\sigma^{(0)} = 1$, $x^{(1)} = 1$;
$\rightarrow$ $s^{(1)} = 0$, $d^{(0)} = -1$, $\sigma^{(1)} = \frac{1}{2}$, $x^{(2)} = \frac{1}{2}$;
...
and finally we get the iteration sequence
$x^{(0)} = 2$, $x^{(1)} = 1$, $x^{(2)} = \frac{1}{2}$, $x^{(3)} = \frac{1}{6}$, $x^{(4)} = -\frac{1}{12}$, $x^{(5)} = \frac{7}{60}$.

ii) Again, we have the sub-differential (2) and the descent directions (3), for the Projected Subgradient Method it follows for $k = 0, 1...5$ and $x^{(0)} = 2$:
$\rightarrow s^{(0)} = -1$, $d^{(0)} = -1$, $\sigma^{(0)} = 2$, $x^{(1)} = 0$;
$\rightarrow s^{(1)} = 0$;
Therefore the problem described above converges after just one iteration with the suitable stepsize $\sigma^{(k)} = \frac{f(x^{(k)} - f(x^*)}{|s^{(k)}|}$.

c) We define

$$g(x) := 2(|x| + 1) \operatorname{sign}(x) \in \partial f(x),$$

a function that assigns each point a possible subgradient of f. Now, we can compute $d^{(k)}$:

$$d^{(k)} = \frac{-g(x^{(k)})}{|g(x^{(k)})|} = \frac{-s^{(k)}}{|s^{(k)}|} = \frac{-2(|x^{(k)}| + 1) \operatorname{sign}(x^{(k)})}{2(|x^{(k)}| + 1)} = -\operatorname{sign}(x^{(k)}).$$

Next, we compute $\sigma^{(k)}$:

$$\begin{aligned} \sigma^{(k)} &= \frac{f(x^{(k)}) - f(x^*)}{|s^{(k)}|} = \frac{(|x^{(k)}| + 1)^2 - 1}{2(|x^{(k)}| + 1)} = \frac{|x^{(k)}|^2 + 2|x^{(k)}|}{2|x^{(k)}| + 2} \\ &= \frac{\frac{1}{2}|x^{(k)}|(2|x^{(k)}| + 2) + |x^{(k)}|}{2|x^{(k)}| + 2} = \frac{|x^{(k)}|}{2} + \frac{|x^{(k)}|}{2|x^{(k)}| + 2}. \end{aligned} \tag{4}$$

An iteration step is defined by

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \sigma^{(k)} d^{(k)} = x^{(k)} - \operatorname{sign}(x^{(k)}) \left(\frac{|x^{(k)}|}{2} + \frac{|x^{(k)}|}{2|x^{(k)}| + 2} \right) \\ &= x^{(k)} \underbrace{\left(1 - \frac{1}{2} - \frac{1}{2|x^{(k)}| + 2} \right)}_{\leq \frac{1}{2}} \leq \frac{1}{2} x^{(k)}. \end{aligned} \tag{5}$$

Thereby it holds that $|x^{(k+1)} - x^*| = |x^{(k+1)}| \leq \frac{1}{2}|x^{(k)}| = \frac{1}{2}|x^{(k)} - x^*|$.

2. *(Projected Subgradient Method)*

a) Apply the routine `projected_subgradient_method.m` to the Wolfe function

$$f^{\text{Wolfe}}(x,y) := \begin{cases} 5\sqrt{9x^2+16y^2}, & x \geq |y|, \\ 9x+16|y|, & 0 < x < |y|, \\ 9x+16|y|-x^9, & x \leq 0 \end{cases}$$

using different step sizes $\sigma^{(k)} = n/(k+1)$, e.g. $n = 1, 2, 3$. Plot the iteration path together with the solution $x^* = (1,0)^T$ and the contour lines of f^{Wolfe}.

b) Write a MATLAB routine

`x = Projection_Parabel(x, a, b),`

which performs a projection on the convex set

$$\mathcal{F} = \{(x,y)^\top \in \mathbb{R}^2 : y \geq (x-a)^2 + b\},$$

where $a, b \in \mathbb{R}$ are two parameters. Solve the constrained optimization problem

$$\min_{(x,y)^T \in \mathcal{F}} f^{\text{Wolfe}}(x,y)$$

for different parameter a and b by using the Projected Subgradient Method. Plot again the iteration path together with the set $\mathcal{F}$ and the contour lines of f^{Wolfe}.

(6 + 6 = 12 Points)

Solution:

a) The script could look like

```
clear, close all
clc

xmax = [-2,6];
ymax = [-2,4];

a = 0;
b = -0.5;

f =@(x) wolfe(x);
x0 = [5;4];
sigma =@(k) 3/(k+1);
subgrad_f =@(x) Subgrad_Wolfe(x);
proj =@(x) x; %Exercise 2b
% proj =@(x) Projection_Parabel(x, a, b); %Exercise 2c
tol   = 1e-2;
maxIt = 1000;
x_sol = [-1;0];
outflag = 0;

[x, f_val, X, iter] = projected_subgradient_method(f, subgrad_f, proj, x0, ...
                                maxIt, sigma, tol,...
                                outflag);
```

```
[XX, YY] = meshgrid(linspace(xmax(1),xmax(2),200), linspace(ymax(1),ymax(2),200));
ZZ = zeros(size(XX));
for i = 1:size(XX,1)
    for j = 1:size(XX,2)
        ZZ(i,j) = wolfe([XX(i,j); YY(i,j)]);
    end
end

figure();
[~,c] = contour(XX,YY,ZZ,150);
hold on;
nb = plot(XX(1,:), (XX(1,:) - a).^2 + b, 'linewidth',3, 'color',[0.8500,0.3250,0.0980]);
p  = plot(X(1,:), X(2,:), '-*', 'linewidth',3,'color',[0, 0.4470, 0.7410]);
so = plot(x_sol(1), x_sol(2), 'ko','linewidth',2, MarkerSize=10);
sg = plot(X(1,end), X(2,end), 'rx','linewidth',2, MarkerSize=10);

xlim(xmax);
ylim(ymax);

tit  = '\textbf{Iterations of the Projected Subgradient Method}';
xlab = '$x$';
ylab = '$y$';
leg  = legend([c,nb,p,so,sg], '\, Wolfe-function', '\, convex set $\mathcal{F}$', ...
                            '\, Subgradient Iterations', ...
                            '\, sol $x^*$', '\, $x^{(k)}$');

title(tit,  'Interpreter','latex')
xlabel(xlab,'Interpreter','latex')
ylabel(ylab,'Interpreter','latex')
set(leg,    'Interpreter','latex','Location','north','FontSize',20);

set(gca,'TickLabelInterpreter', 'latex', 'FontSize',20)
set(gcf,'units','normalized','outerposition',[0 0.05 1 0.9]); % Maximize figure window

% wolfe-function
function fx = wolfe(x)
    if x(1) >= abs(x(2))
        fx = 5*sqrt(9*x(1)^2 + 16*x(2)^2);
    elseif x(1) <= 0
        fx = 9*x(1) + 16*abs(x(2)) - x(1)^9;
    else
        fx = 9*x(1) + 16*abs(x(2));
    end
end
```

b) The function `Projection_Parabel.m` could look like

```
function [x] = Projection_Parabel(x, a, b)

if (x(1) - a)^2 + b <= x(2)
    return
end
p = zeros(4,1);
p(1) = -2;
p(2) = 6*a;
p(3) = 2*x(2) - 1 - 2*b - 6*a^3;
p(4) = x(1) + 2*a*b - 2*x(2)*a + 2*a^3;

r = roots(p);
r = r(imag(r)==0);
```

```
yr = (r - a).^2 + b;

if length(r) ~= 1
    error('something went wrong');
end

x = [r; yr];

end
```