MAT3007 Assignment 6

A6.1

(a)

False. Consider $f : x \mapsto -x$ and $g : x \mapsto x^2$, which are both convex. But $f \circ g : x \mapsto -x^2$ is not.

(b)

True. Let $a, b \in \Omega$ be distinct. Since Ω is convex any convex combination of a, b is also in Ω. It follows that for all $\lambda \in [0,1]$,

$$\begin{aligned}
& g((1-\lambda)a+\lambda b) \le (1-\lambda)g(a)+\lambda g(b) && \text{(convexity of } g\text{)}\\
\implies\ & f\circ g((1-\lambda)a+\lambda b) \le f((1-\lambda)g(a)+\lambda g(b)) && (f \text{ nondecreasing on } I)\\
& \le (1-\lambda)f\circ g(a)+\lambda f\circ g(b). && \text{(convexity of } f\text{)}
\end{aligned}$$

(c)

False. Replacing $\le$ on the second and third lines of (b) with $\ge$, we see that $f \circ g$ is concave then in general. f and g in (a) serve again as a counterexample.

(d)

False. Consider $f : x \mapsto S(x) = \frac{1}{1+\exp(-x)}$, the Sigmoid function. Then f is increasing and non-negative. For $g : \mathbb{R} \to \mathbb{R}$ by $x \mapsto xf(x)$, the derivative is

$$g'(x) = f(x) + xf'(x) = f(x) \cdot [1 + x(1 - f(x))].$$

We have

$$g'(2) = \frac{e^2(3+e^2)}{(1+e^2)^2} \approx 1.09 < \lim_{x \to +\infty} g'(x) = 1.$$

Since g is twice continuously differentiable, it cannot be that $g''(x) \geq 0$ for all positive x. Thus g cannot be convex.

A6.2

(a)

Ω_1 is not convex. Consider $(x_1, t_1) = (0, -1)$ and $(x_2, t_2) = (1, 1)$, which are both in Ω_1 with $n = 1$. But then with $\lambda = 1/2 \in [0, 1]$,

$$(1-\lambda)(x_1, t_1) + \lambda(x_2, t_2) = (\lambda, 2\lambda - 1) = (1/2, 0) \notin \Omega_1.$$

Ω_2 is convex. Let $x_1, x_2 \in \Omega_2$ be distinct. We have

$$\begin{aligned} ||x_1 - a||^2 - ||x_1 - b||^2 &= 2x_1(b-a)^\top + ||a||^2 - ||b||^2 \leq 0 \\ ||x_2 - a||^2 - ||x_2 - b||^2 &= 2x_2(b-a)^\top + ||a||^2 - ||b||^2 \leq 0 \end{aligned}$$

Hence for all $\lambda \in [0, 1]$,

$$\begin{gathered} 2(1-\lambda)x_1(b-a)^\top + (1-\lambda)\left(||a||^2 - ||b||^2\right) \leq 0 \\ 2\lambda x_2(b-a)^\top + \lambda\left(||a||^2 - ||b||^2\right) \leq 0. \end{gathered}$$

Adding, we have

$$2(b-a)^\top((1-\lambda)x_1 + \lambda x_2) + ||a||^2 - ||b||^2 \leq 0. \tag{1}$$

On the other hand,

$$\begin{aligned} &||(1-\lambda)x_1 + \lambda x_2 - a||^2 - ||(1-\lambda)x_1 + \lambda x_2 - b||^2 \\ &= -2a^\top((1-\lambda)x_1 + \lambda x_2) - \left[-2b^\top((1-\lambda)x_1 + \lambda x_2)\right] + ||a||^2 - ||b||^2 \\ &= 2((1-\lambda)x_1 + \lambda x_2)(b^\top - a^\top) + ||a||^2 - ||b||^2 \\ &= 2(b-a)^\top((1-\lambda)x_1 + \lambda x_2) + ||a||^2 - ||b||^2 \\ &\leq 0 \end{aligned}$$

by (1). Hence $(1-\lambda)x_1 + \lambda x_2 \in \Omega_2$, which completes the proof.

(b)

Suppose $x, y \in \mathbb{R}_+^2$ with $x_1 x_2 \geq 1$ and $y_1 y_2 \geq 1$. Then for all $\lambda \in [0, 1]$, the point

$$
\begin{aligned}
(1-\lambda)x+\lambda y &= ((1-\lambda)x_1+\lambda y_1, (1-\lambda)x_2+\lambda y_2) \\
&\geq (\min\{x_1, y_1\}, \min\{x_2, y_2\}) \\
&\geq 0,
\end{aligned}
$$

and

$$
\begin{aligned}
&((1-\lambda)x_1+\lambda y_1)((1-\lambda)x_2+\lambda y_2) \\
&= (1-\lambda)^2 x_1 x_2 + \lambda^2 y_1 y_2 + \lambda(1-\lambda)(x_1 y_2 + y_1 x_2) \\
&\geq (1-\lambda)^2 + \lambda^2 + \lambda(1-\lambda) \cdot 2\sqrt{x_1 y_2 x_2 y_2} \\
&\geq 2\lambda^2 - 2\lambda + 1 + 2\lambda(1-\lambda) \\
&= 1,
\end{aligned}
$$

completing the proof.

(c)

(d)

For the objective function, $-x_1 - x_2$ is linear in x and thus convex. For the max function, suppose we have distinct $u, v \in \mathbb{R}^2$, then for all $\lambda \in [0,1]$,

$$
\begin{aligned}
\max\{(1-\lambda)u+\lambda v\} &= \max\{(1-\lambda)u_1+\lambda v_1, (1-\lambda)u_2+\lambda v_2\} \\
&\leq \max\{(1-\lambda)u_1+\lambda v_1, (1-\lambda)u_2+\lambda v_2, (1-\lambda)u_1+\lambda v_2, (1-\lambda)u_2+\lambda v_1\} \\
&= \max\{(1-\lambda)u_1, (1-\lambda)u_2\} + \max\{\lambda v_1, \lambda v_2\} \\
&= (1-\lambda)\max\{u\} + \lambda \max\{v\}.
\end{aligned}
$$

Hence $\max\{x_3, x_4\}$ is also convex. By Sum Rule the objective function is convex.

For constraints, the last two constraints are linear and thus convex. For the first constraint, since x^2 and x^4 are both convex, by Sum Rule $x_1^2 + x_2^4$ is also convex. Then by Composition with Linear Functions $(x_1 - x_2)^2 + (x_3 + 2x_4)^4$ is a convex function. Thus the first constraint defines a convex level set. Taking intersection of the convex constraints we end up with a convex feasible region. Since the objective function and the feasible region are both convex, the problem is convex by definition.

MATLAB code:

```
cvx_begin
    variable x(4)
    minimize( -x(1) - x(2) + max(x(3),x(4)) )
    subject to
        (x(1)-x(2))^2 + (x(3)+2*x(4))^4 <= 5
        [1 2 1 2] * x <= 6
        x >= zeros(4,1)
cvx_end
```

yielding

$$x^* = (3.490712, 1.254644, 0, 0)$$

with optimal value -4.745356.

A6.3

(a)

MATLAB code:

```
h   = @(x) 2*(x-1)/(x+1);
f   = @(x) ( h(x) - log(x) ) / (x-1)^2;
[x, fx, iter] = ausection(f, 1.5, 4.5, 1e-5);
fprintf("Golden Section: Minimum %f found at x = %f after
%d iterations.\n", fx, x, iter);

function [x, fx, iter] = ausection(f, l, r, e)

    % minimize function f using golden section method.
    % args: f: function handle
    %       l: left endpoint
    %       r: right endpoint
    %       e: max error tolerance

    PHI     = (3 - sqrt(5)) / 2;
    iter    = 0;

    if (l > r)
        fprintf("error: l > r\n")
        return
    end

    nl  = (1 - PHI) * l + PHI * r;
    nr  = (1 - PHI) * r + PHI * l;
    fnl = f(nl);
    fnr = f(nr);

    while (r - l > e)
        iter    = iter + 1;
        if (fnl > fnr)
            l   = nl;
            nl  = nr;
            fnl = fnr;
            nr  = (1 - PHI) * r + PHI * l;
            fnr = f(nr);
        else
            r   = nr;
            nr  = nl;
            fnr = fnl;
            nl  = (1 - PHI) * l + PHI * r;
            fnl = f(nl);
        end
```

```
    end

    x       = (r + l) / 2;
    fx      = f(x);
end
```

```
Golden Section: Minimum -0.026707 found at x = 2.188705
after 27 iterations.
```

(b)

We have

$$g'(x) = -e^{-x} + \sin(x).$$

MATLAB code:

```
g   = @(x) exp(-x) - cos(x);
dg  = @(x) -exp(-x) + sin(x);
au_x, au_fx, au_iter = ausection(g, 0, 1, 1e-5);
bi_x, bi_fx, bi_iter = bisection(g, dg, 0, 1, 1e-5);
fprintf("Golden Section: Minimum %f found at x = %f after
%d iterations.\n", au_fx, au_x, au_iter);
fprintf("Bisection: Minimum %f found at x = %f after %d
iterations.\n", bi_fx, bi_x, bi_iter);

function [x, fx, iter] = bisection(f, df, l, r, e)

    % minimize function f using bisection method.
    % args: f : function handle
    %       df: function derivative
    %       l : left endpoint
    %       r : right endpoint
    %       e : max error tolerance

    iter = 0;

    if (l > r)
        fprintf("error: l > r\n")
        return
    end

    if (df(l) > 0 || df(r) < 0)
        fprintf("error: f not convex")
        return
    end

    while (r - l > e)
        iter    = iter + 1;
        m       = (r + l) / 2;
        if (fp(m) > 0)
```

```
            r = m;
        else
            l = m;
        end
    end

    x       = (r + l) / 2;
    fx      = f(x);
end
```

```
Golden Section: Minimum -0.276615 found at x = 0.588531
after 24 iterations.
Bisection: Minimum -0.276615 found at x = 0.588535 after
17 iterations.
```

We see the bisection method reaches the same accuracy within fewer iterations in this case.

A6.4

(a)

MATLAB code:

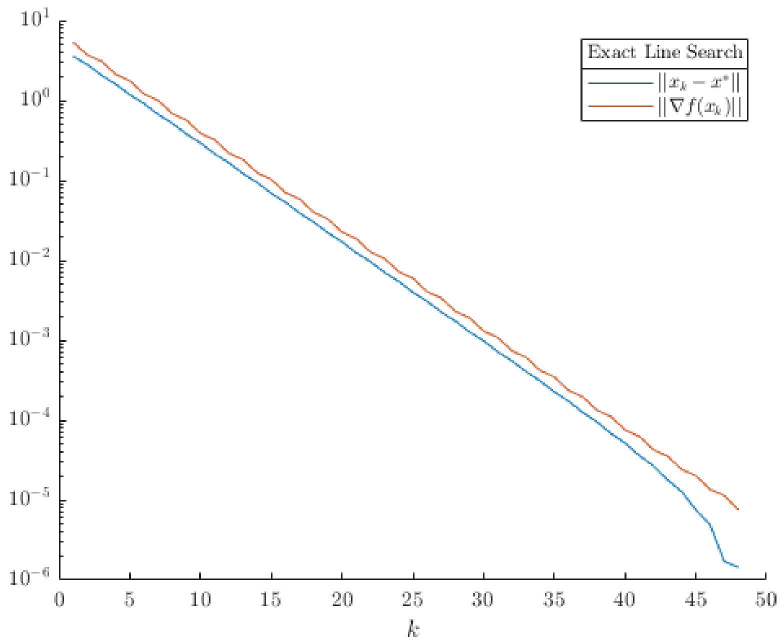
```
e1  = @(x) exp(1-x(1)-x(2));
e2  = @(x) exp(x(1)+x(2)-1);
f   = @(x) e1(x) + e2(x) + x(1)^2 + x(1)*x(2) ...
          + x(2)^2 + 2*x(1) - 3*x(2);
df  = @(x) [-e1(x) + e2(x) + 2*x(1) + x(2) + 2;
            -e1(x) + e2(x) + 2*x(2) + x(1) - 3];

[ex, efx, eiter] = gd(f, df, [0;0], 1e-5, "exact", 1/2,
1/2);
[bx, bfx, biter] = gd(f, df, [0;0], 1e-5, "backtrack",
1/2, 1/2);
fprintf("Exact Line Search: Minimum %f found at x =
(%f,%f) after %d iterations.\n", efx, ex', eiter);
fprintf("Backtracking: Minimum %f found at x = (%f,%f)
after %d iterations.\n", bfx, bx', biter);

function [x, fx, iter] = gd(f, df, init, tol, line_search,
sigma, gamma)

    % minimize function f using gradient descent.
    % args: f: function handle
    %       df: function gradient
    %       init: initial point
    %       tol: stopping tolerance
    %       line_search: "exact" or "backtrack"
```

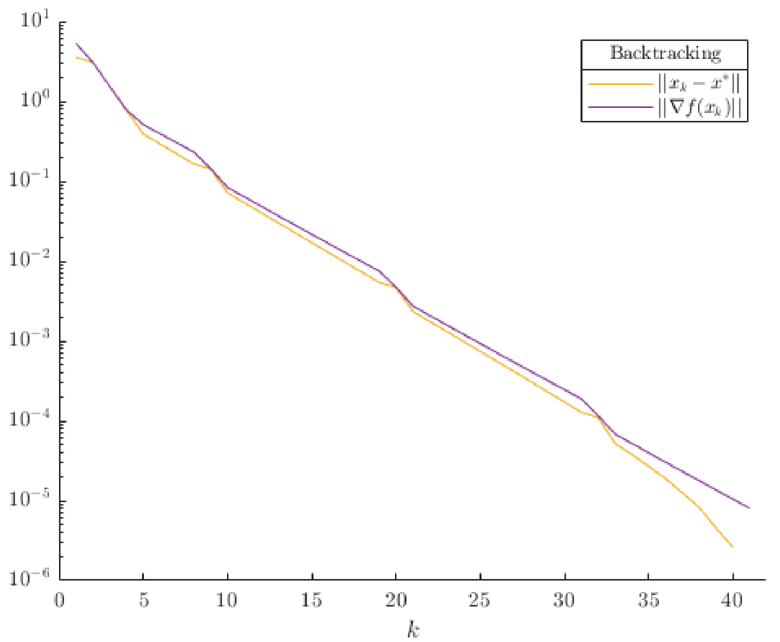
```
    %       sigma, gamma: backtracking parameters

    iter    = 1;
    x       = init;
    fx      = f(x);
    dfx     = df(x);
    nrm     = norm(dfx);

    while (nrm > tol)
        fprintf("iter:%02d      x:(%f,%f)     norm:%f
optval:%f\n", ...
                iter, x(1), x(2), nrm, fx);

        if (line_search == "exact")
            step = ausection(@(a) f(x-a*dfx), 0, 10, 1e-
5);

        elseif (line_search == "backtrack")
            step = 1;
            while ( f(x-step*dfx) - f(x) > -
gamma*step*nrm^2 )
                step = step * sigma;
            end

        else
            fprintf("invalid line search method");
            return
        end

        x    = x - step * dfx;
        fx   = f(x);
        dfx  = df(x);
        nrm  = norm(dfx);
        iter = iter + 1
    end

end
```

```
Exact Line Search: Minimum -4.142309 found at x =
(-2.141763,2.858229) after 48 iterations.
Backtracking: Minimum -4.142309 found at x =
(-2.141764,2.858228) after 41 iterations.
```

We see the backtracking line search finds the minimum within fewer iterations.

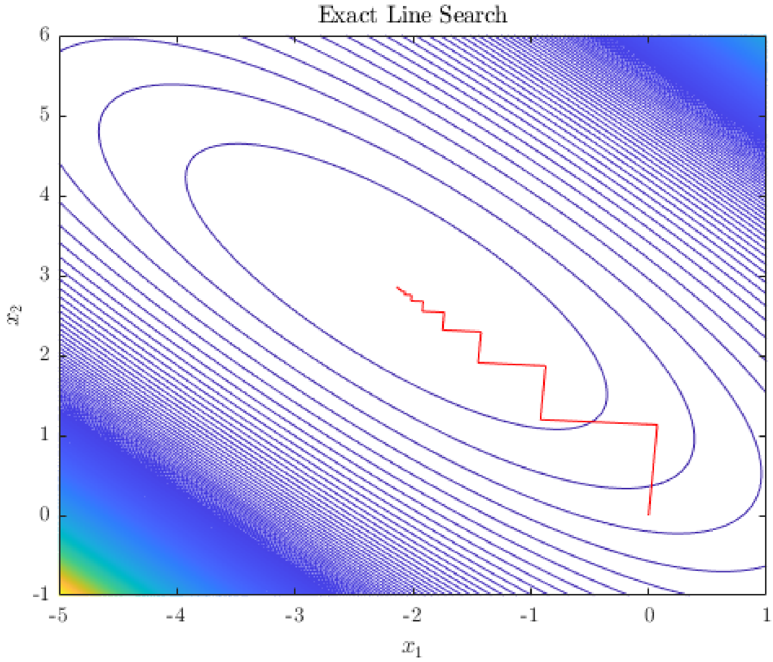
(b)

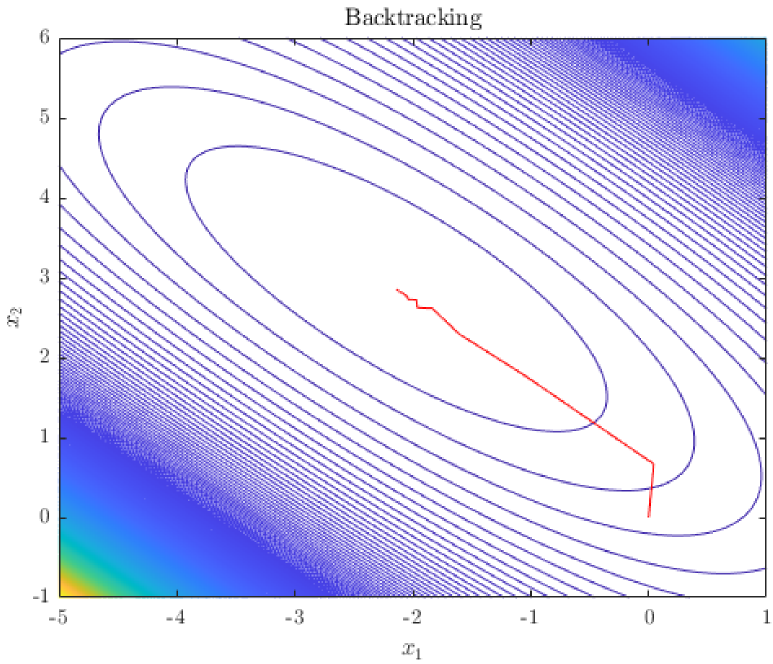




Both methods converge exponentially.

(c)


Exact Line Search
6
5
4
3
2
1
0
-1
x_2
-5
-4
-3
-2
-1
0
1
x_1


Backtracking
6
5
4
3
2
1
0
-1
x_2
-5
-4
-3
-2
-1
0
1
x_1