EE 236B HW4
Zhiyuan Cao
304397496
02/09/2017

T3.55

(a)

$$f'(x) = e^{-h(x)} \qquad\qquad f''(x) = -h'(x)e^{-h(x)}$$

It is clear we have,

$$(f'(x))^2 \geq 0 \geq -h'(x)e^{-2h(x)} = f''(x)f(x)$$

(b)

Take exponential and integrate both side of $e^{-h(t)} \leq e^{-h(x)-h'(x)(t-x)}$

$$\begin{aligned}
\int_{-\infty}^{x} e^{-h(t)}dt &\leq e^{xh'(x)-h(x)} \int_{-\infty}^{x} e^{-h'(x)t}dt \\
&= e^{xh'(x)-h(x)} \frac{e^{-xh'(x)}}{-h'(x)} \\
&= \frac{e^{-h(x)}}{-h'(x)}
\end{aligned}$$

From here we have $-h'(x)e^{-h(x)} \int_{-\infty}^{x} e^{-h(t)}dt \leq e^{-2h(x)}$ which is just $f''(x)f(x) \leq (f'(x))^2$.

A3.5

(a)

Name the original problem as (1) and first introduce t and s to acquire the following problem as (2)

$$\begin{array}{ll}
\textit{minimize} & t/s \\
\textit{subject to} & \max\limits_{i=1,\ldots,m} a_i^T x + b_i \leq t \\
& \min\limits_{i=1,\ldots,p} c_i^T x + d_i \geq s \\
& Fx \preceq g \\
& s \geq 0
\end{array}$$

It is obvious that (1) and (2) are equivalent since they have the same feasible set and notice that (2) reaches optimality with the following two equations $\max_{i=1,...,m} a_i^T x + b_i = t$ and $\min_{i=1,...,p} c_i^T x + d_i = s$, i.e., they have the same optimal value. Now this is a linear-fractional programming problem with $s > 0$. We can apply the same trick used in textbook 4.3.2 and rewrite the problem as (3):

$$
\begin{array}{lll}
\textit{minimize} & u & \\
\textit{subject to} & a_i^T x + b_i \leq u & i = 1, ..., m \\
 & c_i^T x + d_i \geq v & i = 1, ..., p \\
 & Fx \preceq g & \\
 & v = 1 &
\end{array}
$$

This is now an linear programming problem. To see equivalent, for a feasible u we set $t = u/v$ and $s = 1/v$, which is feasible in (2) with the same objective value. Conversely, if t, s is feasible in (2), then $u = t/s$ is feasible in (3) with same objective value.

T4.21

(b)

Let $y = A^{1/2}(x - x_c)$ and $\hat{c} = A^{-1/2}c$, the original problem could be written as:

$$
\begin{array}{ll}
\text{minimize} & \hat{c}^T y + \hat{c}^T x_c \\
\text{subject to} & y^T y \leq 1
\end{array}
$$

The solution to this problem can be easily seen geometrically, y lies on the boundary of the unit ball and has the opposite direction of $\hat{c}$, i.e., $y = -\hat{c}/||\hat{c}||$. The corresponding solution to x therefore is

$$
x = x_c - \frac{cA^{-1}}{\sqrt{c^T A^{-1} c}}
$$

T4.25

First we can replace the two constraints with $a^T x + b \geq 1$ and $a^T x + b \leq -1$ due to homogeneity. From here we have the following constraints:

$$
\begin{array}{ll}
\inf_{\varepsilon_i} \{a^T x + b\} \geq 1 & i = 1, ..., K \\
\sup_{\varepsilon_i} \{a^T x + b\} \leq -1 & i = K + 1, ..., K + L
\end{array}
$$

Plugging in the definition of ellipsoid we have

$$
\begin{aligned}
\inf_{\varepsilon_i}\{a^T x + b\} &= \inf_{\varepsilon_i}\{a^T q_i + a^T P_i u + b \mid ||u||_2 \leq 1\} \\
&= a^T q_i - ||P_i^T a||_2 + b \\
\sup_{\varepsilon_i}\{a^T x + b\} &= a^T q_i + ||P_i^T a||_2 + b
\end{aligned}
$$

This is followed by Cauchy Inequality. Therefore we acquired two groups of SOCP constraints and the problem can be formed as follow:

$$
\begin{array}{lll}
find & a, b & \\
subject\ to & a^T q_i - ||P_i^T a||_2 + b \geq 1, & i = 1, ..., K \\
 & a^T q_i + ||P_i^T a||_2 + b \leq -1 & i = K+1, ..., K+L
\end{array}
$$

A7.9

(a)

Introducing $\phi_{k,t}(x) = ||A_k x + b_k - y_k(c_k^T x + d_k)||_2 - t(c_k^T x + d_k)$ and we can solve a feasibility problem of finding x with $\phi_{k,t}(x) \leq 0$. The original problem is a quasiconvex problem since each $f_k(x) - y_k$ is quasilinear and taking the max over a family of them does not change convexity.

(b)

With CVX the solution is found at $t = 0.495$ and $x = [4.9\ 5.0\ 5.2]$.

```
clc;clear;
P1 = [1, 0, 0, 0; 0, 1, 0, 0; 0, 0, 1, 0];
P2 = [1, 0, 0, 0; 0, 0, 1, 0; 0, -1, 0, 10];
P3 = [1, 1, 1, -10; -1, 1, 1, 0; -1, -1, 1, 10];
P4 = [0, 1, 1, 0; 0, -1, 1, 0; -1, 0, 0, 10];
y1 = [0.98; 0.93];
y2 = [1.01; 1.01];
y3 = [0.95; 1.05];
y4 = [2.04; 0.00];
a = 0;
b = 1;
t = (a+b)/2;
tol = 1e-4;
while b-a > tol
    cvx_begin
        variable x(3)
        subject to
```

```
            norm(P1(1:2, 1:3)*x + P1(1:2, 4) - (P1(3, 1:3)*x + P1(3, 4))*y1, 2)...
                - t*(P1(3, 1:3)*x + P1(3, 4)) <= 0
            norm(P2(1:2, 1:3)*x + P2(1:2, 4) - (P2(3, 1:3)*x + P2(3, 4))*y2, 2)...
                - t*(P2(3, 1:3)*x + P2(3, 4)) <= 0
            norm(P3(1:2, 1:3)*x + P3(1:2, 4) - (P3(3, 1:3)*x + P3(3, 4))*y3, 2)...
                - t*(P3(3, 1:3)*x + P3(3, 4)) <= 0
            norm(P4(1:2, 1:3)*x + P4(1:2, 4) - (P4(3, 1:3)*x + P4(3, 4))*y4, 2)...
                - t*(P4(3, 1:3)*x + P4(3, 4)) <= 0
    cvx_end
    if cvx_optval == Inf
        a = t;
    else
        b = t;
    end
    t = (a+b)/2;
end
```

A14.8

(a)

The problem can be formed into the following optimization problem:

$$\begin{array}{ll}
\textit{minimize} & \sum_{K} ||f_k(t)||_2 \\
\textit{subject to} & ||f_k(t)||_2 \leq \text{Fmax} \qquad k = 1, ..., K \\
& \alpha ||A^T p_k(t)||_2 - e_3 p_k(t) \leq 0 \\
& v_{k+1} = v_k + (h/m) f_k - hge_3 \\
& p_{k+1} = p_k + (h/2)(v_k + v_{k+1}) \\
& v_{des} = p_{des} = 0 \\
& p_{init} = p_0 \\
& v_{init} = v_0
\end{array}$$

Where variable $[f_k(t), p_k(t), v_k(t)]^T \in R^{9\times 36}$, and $A = [1\ 1\ 0]^T$. This is a convex optimization problem since the objective function is convex and all constraints are either convex or linear.

(b)

We can solve this problem by fixing K to a number starting from 1 and increasing K till we find a solution.

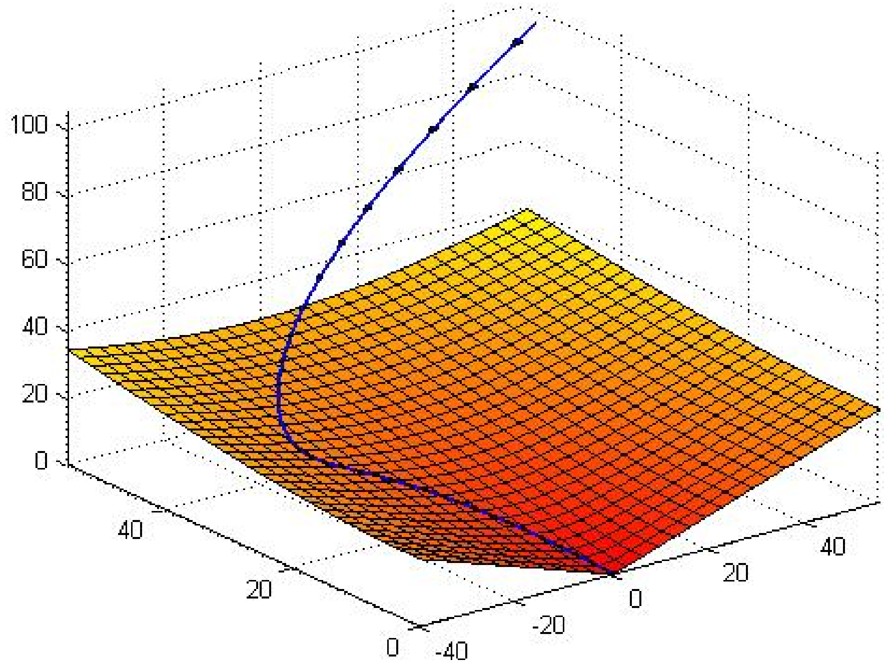

Figure 1: Min Fuel Trajectory

(c)

For (a) $p^\star = 193$ and for (b) $p^\star = 25$. Attached are code and figures.

Min Fuel

```
clc;clear;
spacecraft_landing_data;
K =25;
e33 = [zeros(2, K);ones(1, K)];
F =Fmax *ones(1, K+1);
cvx_begin
    variable x(9, K+1)
    minimize sum(norms(x(1:3,:)))
    subject to
        norms(x(1:3,:)) <= F
        alpha*norms(x(4:5,:)) - x(6,:) <=0
        x(7:9, 2:K+1) == x(7:9, 1:K) + (h/m)*x(1:3, 1:K) - h*g*e33
        x(4:6, 2:K+1) == x(4:6, 1:K) + (h/2)*(x(7:9, 2:K+1) + x(7:9, 1:K))
        x(4:6, 1) == p0
        x(7:9, 1) == v0
        x(4:9, K+1) == 0
```

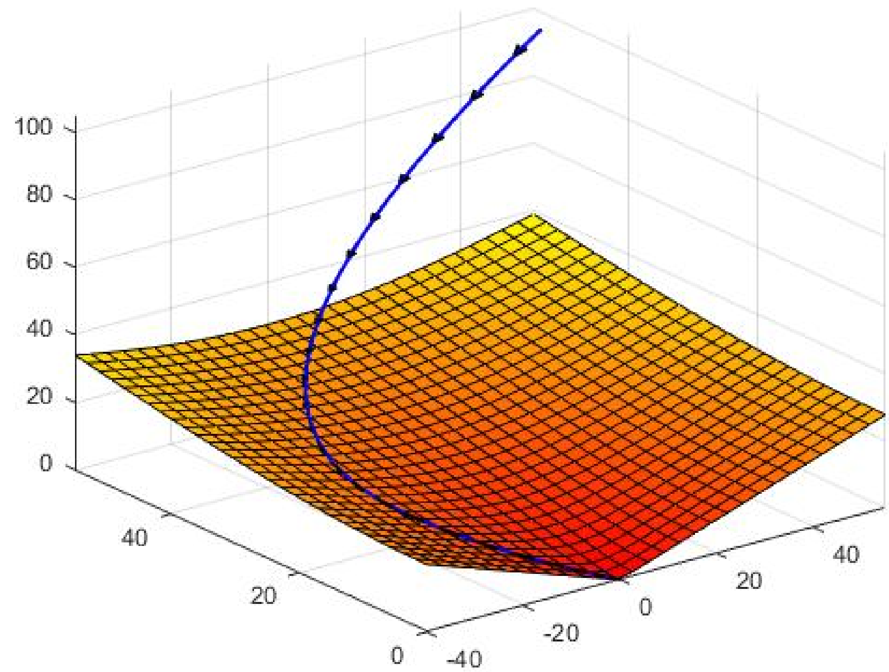

Figure 2: Min Time Trajectory

```
cvx_end
```

Min Time

```
clc;clear;
spacecraft_landing_data;
for i = 1:35
   K = i;
   sc_ld_main; % Main function block that solve the problem once
   clc;
   if cvx_optval ~= Inf
       break;
   end
end
```