Convex Optimization Homework #3 ,

Due: Saturday June 26, 2021, 11am.

r09942082 楊雅婷

1. (110%) In this problem, we aim to solve the unconstrained problem

$$\underset{\tilde{x}\in\mathbf{R}^2}{\text{minimize}} \quad \max_k f_k(\tilde{x}) \tag{1}$$

where $f_k(\tilde{x}) = \frac{1}{2}(\tilde{x}-y_k)^T P_k(\tilde{x}-y_k) + r_k$ with

$$P_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},\ P_2 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix},\ P_3 = \begin{bmatrix} 2 & -1 \\ -1 & 3 \end{bmatrix}, y_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, y_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, y_3 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}, r_1 = 0, r_2 = 1, r_3 = -1.$$

The objective function of Problem (1) is not differentiable, and therefore the Newton's method cannot be applied directly. Fortunately, the problem can be transformed into an equivalent problem in QCQP:

$$\begin{aligned} \underset{\tilde{x}\in\mathbf{R}^2, w\in\mathbf{R}}{\text{minimize}} \quad & w \\ \text{subject to} \quad & f_1(\tilde{x}) - w \le 0 \\ & f_2(\tilde{x}) - w \le 0 \\ & f_3(\tilde{x}) - w \le 0, \end{aligned} \tag{2}$$

whose objective function and constraint functions are all convex and twice differentiable. We will apply the barrier method we learned in class to solve Problem (2).

(a) (10%) Let $x = [\ \tilde{x}^T, w\ ]^T = [\ x_1, x_2, x_3\ ]^T$, $f_0(x) = w = x_3$, and $\phi(x) = -\sum_{i=1}^m \log(-f_i(\tilde{x}) + w)$. Let $f(x) = tf_0(x) + \phi(x)$ for any $t > 0$. Find **dom** f and derive $\nabla f(x)$ and $\nabla^2 f(x)$.

(b) (5%) Write a matlab function which takes inputs x, and t, and evaluates the function f at the point x, as well as the gradient and the Hessian.
Hint: the function can have a header that looks like the following.

```
function [f,g,H] = my_objective_with_log_barrier(x, t)
```

Then, prepare to write an m-file as the main script for your program to solve Problem (2).

(c) (20% in total for (c)-(k)) Set the initial point $x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}$ and let $t = 1$. Let $l = 0$ denoting the Newton iteration index.

(d) Use the Newton step's formula: $\Delta x_{\text{nt}} = -(\nabla^2 f(x))^{-1}\nabla f(x)$ and calculate the Newton step $\Delta x_{\text{nt}}^{(l)}$.

(e) Calculate the Newton decrement $\lambda^{(l)}(x) = (-\nabla f(x^{(l)})^T \Delta x_{\text{nt}}^{(l)})^{1/2}$.

(f) Perform backtracking line search along search direction using $\beta = 0.7$ starting from $s = 1$ until $x^{(0)} + s\Delta x_{\text{nt}}^{(0)} \in$ **dom** f. (Note: $s^+ := \beta s$) [1]

(g) Continue the backtracking line search until $f(x^{(l)} + s\Delta x_{\text{nt}}^{(l)}) \le f(x^{(l)}) - \alpha s \lambda(x^{(l)})^2$, where $\alpha = 0.1$.

(h) Perform the update $x^{(l+1)} = x^{(l)} + s\Delta x_{\text{nt}}^{(l)}$. Determine whether $\lambda^2/2 \le \epsilon_{\text{inner}}$ where $\epsilon_{\text{inner}} = 10^{-5}$.

(i) Let $l = l + 1$ and repeat (d)-(h) for the next Newton iteration. Wrap up these steps in an inner loop. For the lth iteration, record the following items for future use: (1) $f(x^{(l)})$, (2) $\lambda(x^{(l)})$, (3) $s^{(l)}$. Let $l := l + 1$ at the end of each Newton iteration until $\lambda^2/2 \le \epsilon_{\text{inner}}$ as in (h).

(j) Write an outer loop that include 1) steps of the inner loop obtained from (d) to (i); 2) Updating $t^+ := \mu t$ where $\mu = 20$; 3) Checking if the stopping criterion $m/t \le \epsilon_{\text{outer}}$ is met, where $\epsilon_{\text{outer}} = 10^{-8}$. The Newton iteration index l continues to grow and does not reset to 0 in the event of a new outer iteration.

(k) Run your code, and record the following items: 1) the number of outer iterations; 2) for each outer iteration, the number of inner Newton steps; 3) for each outer iteration, record t and the function value $f(x)$ at the end of the outer iteration. 4) the total number of Newton steps; 5) The optimal point $x^*(t)$ obtained at the end of the last outer iteration.

(l) (5%) Make a comment on the relationship between the boundary and the central path. In particular, determine the number of inequality constraints that are active according to ~~the plot you generated~~. changed,

(m) (60%) Repeat (c)-(k) but replace μ as $\mu = 200, 2000$, and 20000, respectively.

(n) (5%) Compare your results (i.e., optimal value, optimal point) with those you obtained with $\mu = 20$.

[1] Here we choose to use the letter s as the line search parameter instead of t, with the consideration to avoid confusion between the parameter t associated with the logarithm barrier function.

(o) (5%) Which μ among 20, 200, 2000, and 20000, used the lowest total number of Newton iterations?

(p) (10%) Use the `cvx` toolbox to solve the same problem (Problem (2)). Compare your results (i.e., optimal value, optimal point) with those you obtain from `cvx`.

Homework submission guidelines:

- Submit your answer online as a set of three files: two m-files and a document file (in *.pdf or *.docx) that contains all answers (including plots) in this problem set.
- If you choose to do the homework in Python, then submit *.py files instead of m-files.
- Submit your files online at the NTU Cool website. No paper shall be handed in.
- Late submissions will be treated under the principle elaborated as follows.
  (1) Homework received by 11am, Saturday June 26 (t_1) will be counted fully.
  (2) Homework received after 2pm, Saturday June 26 (t_2) will not be counted.
  (3) Homework received between t_1 and t_2 will be counted with a discount rate
  $$\frac{t_2 - t}{t_2 - t_1}$$
  where t is the received time. Note that $t_2 - t_1$ is three hours.
- Plagiarism is strongly prohibited. While discussions among classmates are allowed (and encouraged), you shall not ask anyone else to share his/her codes with you, nor should you attempt to share with anyone your codes. You should write every line of the code by yourself. If any part of your submission is found to be copied from someone else's submission, then both of your homework submissions will be counted zero.

(a)

$$f_1(\tilde{x}) = \frac{1}{2}\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}-\begin{bmatrix}1\\0\end{bmatrix}\right)^T\begin{bmatrix}2&1\\1&2\end{bmatrix}\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}-\begin{bmatrix}1\\0\end{bmatrix}\right)+0$$
$$= \frac{1}{2}\begin{bmatrix}x_1-1 & x_2\end{bmatrix}\begin{bmatrix}2&1\\1&2\end{bmatrix}\begin{bmatrix}x_1-1\\x_2\end{bmatrix}$$
$$= \frac{1}{2}\begin{bmatrix}2(x_1-1)+x_2 & (x_1-1)+2x_2\end{bmatrix}\begin{bmatrix}x_1-1\\x_2\end{bmatrix}$$
$$= \frac{1}{2}\left[2(x_1-1)^2+2(x_1-1)x_2+2x_2^2\right]$$
$$= (x_1-1)^2+(x_1-1)x_2+x_2^2$$

$$f_2(\tilde{x}) = \frac{1}{2}\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}-\begin{bmatrix}-1\\2\end{bmatrix}\right)^T\begin{bmatrix}2&0\\0&3\end{bmatrix}\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}-\begin{bmatrix}-1\\2\end{bmatrix}\right)+1$$
$$= \frac{1}{2}\left(\begin{bmatrix}x_1+1 & x_2-2\end{bmatrix}\begin{bmatrix}2&0\\0&3\end{bmatrix}\begin{bmatrix}x_1+1\\x_2-2\end{bmatrix}\right)+1$$
$$= \frac{1}{2}\left(\begin{bmatrix}2(x_1+1) & 3(x_2-2)\end{bmatrix}\begin{bmatrix}x_1+1\\x_2-2\end{bmatrix}\right)+1$$
$$= \frac{1}{2}\left(2(x_1+1)^2+3(x_2-2)^2\right)+1$$
$$= (x_1+1)^2+1.5(x_2-2)^2+1$$

$$f_3(\hat{x}) = \frac{1}{2}\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}-\begin{bmatrix}-1\\-2\end{bmatrix}\right)^T\begin{bmatrix}2&-1\\-1&3\end{bmatrix}\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}-\begin{bmatrix}-1\\-2\end{bmatrix}\right)-1$$
$$= \frac{1}{2}\left(\begin{bmatrix}x_1+1 & x_2+2\end{bmatrix}\begin{bmatrix}2&-1\\-1&3\end{bmatrix}\begin{bmatrix}x_1+1\\x_2+2\end{bmatrix}\right)-1$$
$$= \frac{1}{2}\left(\begin{bmatrix}2(x_1+1)-(x_2+2) & -(x_1+1)+3(x_2+2)\end{bmatrix}\begin{bmatrix}x_1+1\\x_2+2\end{bmatrix}\right)-1$$
$$= \frac{1}{2}\left(2(x_1+1)^2-(x_1+1)(x_2+2) \quad -(x_1+1)(x_2+2)+3(x_2+2)^2\right)-1$$
$$= (x_1+1)^2-(x_1+1)(x_2+2)+1.5(x_2+2)^2-1$$

$$f_0(x) = w = x_3$$
$$\phi(x) = -\sum_{i=1}^{3}\log(-f_i(\tilde{x})+w)$$
$$\Rightarrow f(x) = tf_0(x)+\phi(x) = tf_0(x)-\sum_{i=1}^{3}\log(-f_i(\tilde{x})+w)$$
$$= tx_3 - \log\left(-\left((x_1-1)^2+(x_1-1)x_2+x_2^2\right)+x_3\right)$$
$$- \log\left(-\left((x_1+1)^2+1.5(x_2-2)^2\right)+1\right)+x_3\right)$$
$$- \log\left(-\left((x_1+1)^2-(x_1+1)(x_2+2)+1.5(x_2+2)^2-1\right)+x_3\right)$$

$x = [x_1 \;\; x_2 \;\; x_3]^T$

$-f_i(\tilde{x})+w = -f_i(x)$

$\Rightarrow f_i(x) = f_i(\tilde{x})-w$

$f_1(x) = (x_1-1)^2+(x_1-1)x_2+x_2^2-x_3$

$f_2(x) = (x_1+1)^2+1.5(x_2-2)^2-x_3+1$

$f_3(x) = (x_1+1)^2-(x_1+1)(x_2+2)+1.5(x_2+2)^2-x_3-1$

$$\mathrm{dom} f = \{x \in \mathbb{R}^3 \mid -f_i(\tilde{x})+w > 0,\ i=1,2,\dots,m\}$$

$$\mathrm{dom} f = \{x \in \mathbb{R}^3 \mid -f_i(x) > 0,\ i=1,2,\dots,m\}$$

$$\Rightarrow \mathrm{dom} f = \{x \in \mathbb{R}^3 \mid x_3-(x_1-1)^2-(x_1-1)x_2-x_2^2 > 0,$$
$$x_3-(x_1+1)^2-1.5(x_2-2)^2-1 > 0,$$
$$x_3-(x_1+1)^2+(x_1+1)(x_2+2)-1.5(x_2+2)^2+1 > 0\}$$

$$\nabla f(x) = \begin{bmatrix}0\\0\\t\end{bmatrix} - \begin{bmatrix}\frac{2(x_1-1)+x_2}{(x_1-1)^2+(x_1-1)x_2+x_2^2-x_3}\\ \frac{(x_1-1)+2x_2}{(x_1-1)^2+(x_1-1)x_2+x_2^2-x_3}\\ \frac{-1}{(x_1-1)^2+(x_1-1)x_2+x_2^2-x_3}\end{bmatrix} - \begin{bmatrix}\frac{2(x_1+1)}{(x_1+1)^2+1.5(x_2-2)^2-x_3+1}\\ \frac{3(x_2-2)}{(x_1+1)^2+1.5(x_2-2)^2-x_3+1}\\ \frac{-1}{(x_1+1)^2+1.5(x_2-2)^2-x_3+1}\end{bmatrix} - \begin{bmatrix}\frac{2(x_1+1)-(x_3+2)}{(x_1+1)^2-(x_1+1)(x_2+2)+1.5(x_2+2)^2-x_3-1}\\ \frac{-(x_1+1)+3(x_2+2)}{(x_1+1)^2-(x_1+1)(x_2+2)+1.5(x_2+2)^2-x_3-1}\\ \frac{-1}{(x_1+1)^2-(x_1+1)(x_2+2)+1.5(x_2+2)^2-x_3-1}\end{bmatrix}$$

$$\nabla f(x) = \begin{bmatrix}0\\0\\t\end{bmatrix} - \sum_{i=1}^{m}\frac{1}{f_i(x)}\nabla f_i(x)$$

$$\nabla^2 f(x) = \sum_{i=1}^{m}\frac{1}{f_i(x)^2}\nabla f_i(x)\nabla f_i(x)^T - \sum_{i=1}^{m}\frac{1}{f_i(x)}\nabla^2 f_i(x)$$

$$\Rightarrow \nabla^2 f(x) = \frac{1}{\left[(x_1-1)^2+(x_1-1)x_2+x_2^2-x_3\right]^2}\begin{bmatrix}2(x_1-1)+x_2\\(x_1-1)+2x_2\\-1\end{bmatrix}\begin{bmatrix}2(x_1-1)+x_2 & (x_1-1)+2x_2 & -1\end{bmatrix} - \frac{\begin{bmatrix}2&1&0\\1&2&0\\0&0&0\end{bmatrix}}{(x_1-1)^2+(x_1-1)x_2+x_2^2-x_3}$$
$$+ \frac{1}{\left[(x_1+1)^2+1.5(x_2-2)^2-x_3+1\right]^2}\begin{bmatrix}2(x_1+1)\\3(x_2-2)\\-1\end{bmatrix}\begin{bmatrix}2(x_1+1) & 3(x_2-2) & -1\end{bmatrix} - \frac{\begin{bmatrix}2&0&0\\0&3&0\\0&0&0\end{bmatrix}}{(x_1+1)^2+1.5(x_2-2)^2-x_3+1}$$
$$+ \frac{1}{\left[(x_1+1)^2-(x_1+1)(x_2+2)+1.5(x_2+2)^2-x_3-1\right]^2}\begin{bmatrix}2(x_1+1)-(x_2+2)\\-(x_1+1)+3(x_2+2)\\-1\end{bmatrix}\begin{bmatrix}2(x_1+1)-(x_2+2) & -(x_1+1)+3(x_2+2) & -1\end{bmatrix} - \frac{\begin{bmatrix}2&-1&0\\-1&3&0\\0&0&0\end{bmatrix}}{(x_1+1)^2-(x_1+1)(x_2+2)+1.5(x_2+2)^2-x_3-1}$$

r09942082
楊雅婷

(b) my_objective_with_log_barrier (x,t) in hw3_objective.py

(c) - (k) in barrier_method (x,u) and solver( ) in hw3_main.py

(l) Outer loop 1 [x_1 x_2 x_3] = [-0.27179211, 0.28064711, 8.16854631]

Outer loop 2 [x_1 x_2 x_3] = [-0.46820907, 0.26551839, 5.88820386]

Outer loop 3 [x_1 x_2 x_3] = [-0.4857604, 0.26359097, 5.79169497]

Outer loop 4 [x_1 x_2 x_3] = [-0.4866753, 0.26348813, 5.78694093]

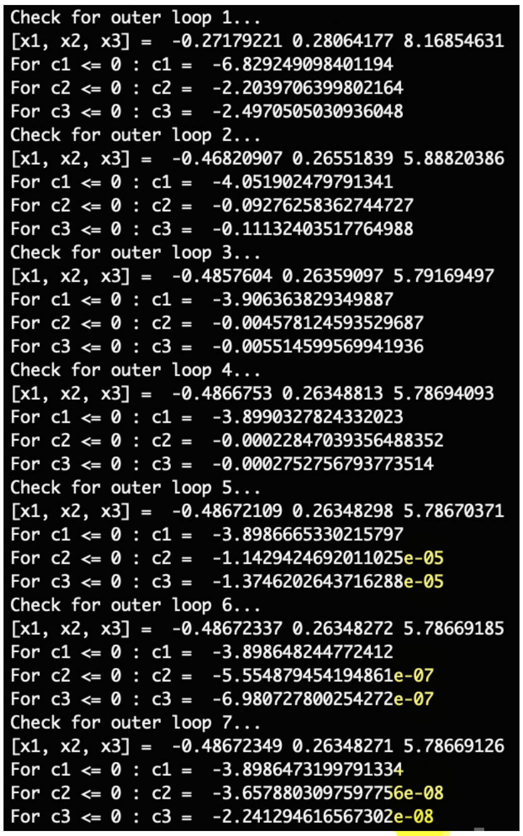
Outer loop 5 [x_1 x_2 x_3] = [-0.48672109, 0.26348298, 5.78670371]

Outer loop 6 [x_1 x_2 x_3] = [-0.48672337, 0.26348272, 5.78669185]

Outer loop 7 [x_1 x_2 x_3] = [-0.48672349, 0.26348271, 5.78669126]

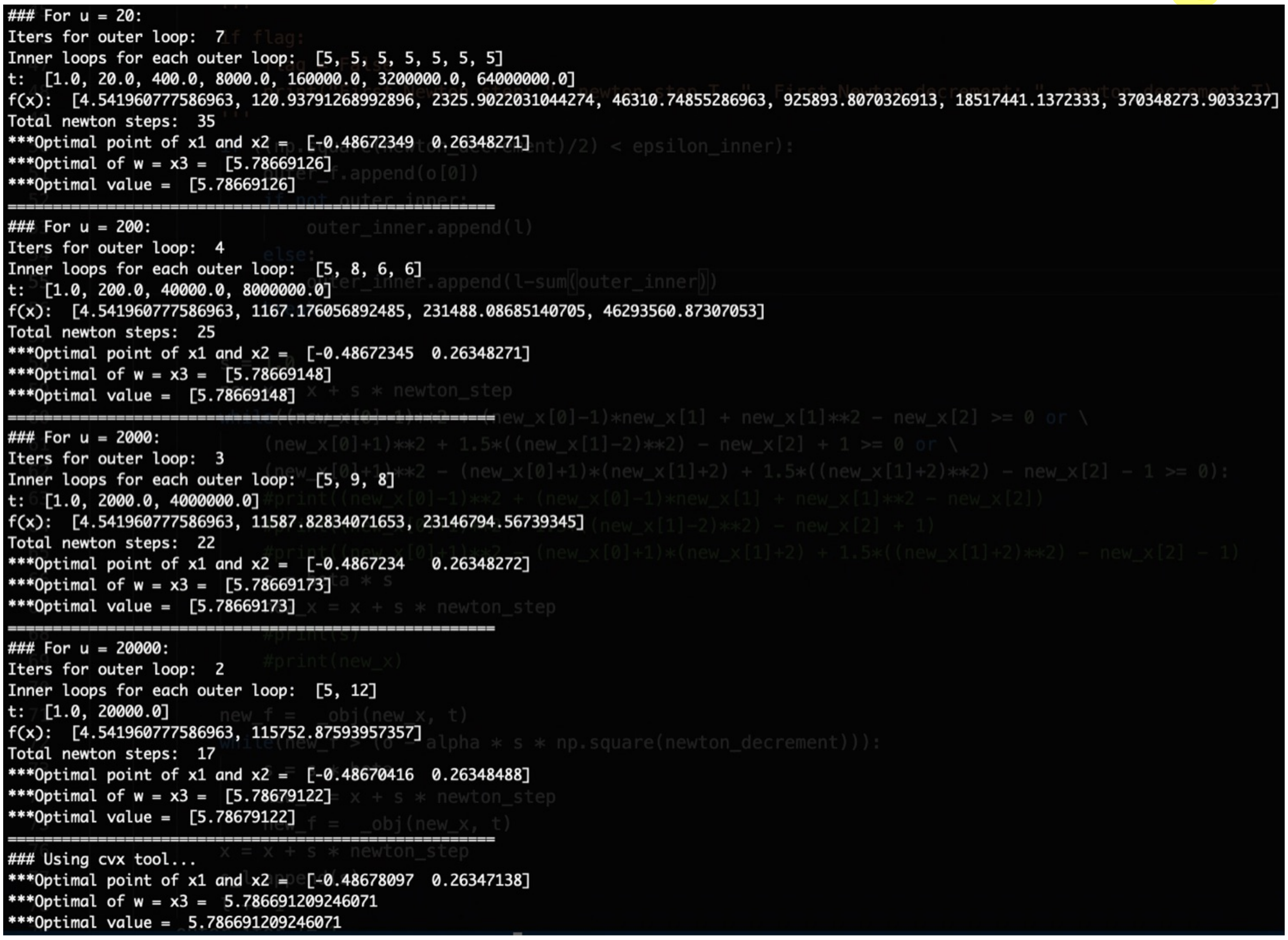
(以 10^{-4} 為界)

# of active	outer	$f_1(\tilde{x})+w \le 0$	$f_2(\tilde{x})+w \le 0$	$f_3(\tilde{x})+w \le 0$
0	1	inactive	inactive	inactive
0	2	inactive	inactive	inactive
0	3	inactive	inactive	inactive
0	4	inactive	inactive	inactive
2	5	inactive	active	active
2	6	inactive	active	active
2	7	inactive	active	active



```
Check for outer loop 1...
[x1, x2, x3] =  -0.27179221 0.28064177 8.16854631
For c1 <= 0 : c1 =  -6.829249098401194
For c2 <= 0 : c2 =  -2.2039706399802164
For c3 <= 0 : c3 =  -2.4970505030936048
Check for outer loop 2...
[x1, x2, x3] =  -0.46820907 0.26551839 5.88820386
For c1 <= 0 : c1 =  -4.051902479791341
For c2 <= 0 : c2 =  -0.09276258362744727
For c3 <= 0 : c3 =  -0.11132403517764988
Check for outer loop 3...
[x1, x2, x3] =  -0.4857604 0.26359097 5.79169497
For c1 <= 0 : c1 =  -3.906363829349887
For c2 <= 0 : c2 =  -0.004578124593529687
For c3 <= 0 : c3 =  -0.005514599569941936
Check for outer loop 4...
[x1, x2, x3] =  -0.4866753 0.26348813 5.78694093
For c1 <= 0 : c1 =  -3.8990327824332023
For c2 <= 0 : c2 =  -0.00022847039356488352
For c3 <= 0 : c3 =  -0.00027527567937735l4
Check for outer loop 5...
[x1, x2, x3] =  -0.48672109 0.26348298 5.78670371
For c1 <= 0 : c1 =  -3.8986665330215797
For c2 <= 0 : c2 =  -1.142942469201l025e-05
For c3 <= 0 : c3 =  -1.3746202643716288e-05
Check for outer loop 6...
[x1, x2, x3] =  -0.48672337 0.26348272 5.78669185
For c1 <= 0 : c1 =  -3.898648244772412
For c2 <= 0 : c2 =  -5.554879454194861e-07
For c3 <= 0 : c3 =  -6.980727800254272e-07
Check for outer loop 7...
[x1, x2, x3] =  -0.48672349 0.26348271 5.78669126
For c1 <= 0 : c1 =  -3.8986473199791334
For c2 <= 0 : c2 =  -3.657880309759775 6e-08
For c3 <= 0 : c3 =  -2.241294616567302e-08
```

(m)



```
### For u = 20:
Iters for outer loop:  7
Inner loops for each outer loop:  [5, 5, 5, 5, 5, 5, 5]
t:  [1.0, 20.0, 400.0, 8000.0, 160000.0, 3200000.0, 64000000.0]
f(x):  [4.541960777586963, 120.93791268992896, 2325.9022031044274, 46310.74855286963, 925893.8070326913, 18517441.1372333, 370348273.9033237]
Total newton steps:  35
***Optimal point of x1 and x2 =  [-0.48672349  0.26348271]
***Optimal of w = x3 =  [5.78669126]
***Optimal value =  [5.78669126]
=================================================
### For u = 200:
Iters for outer loop:  4
Inner loops for each outer loop:  [5, 8, 6, 6]
t:  [1.0, 200.0, 40000.0, 8000000.0]
f(x):  [4.541960777586963, 1167.176056892485, 231488.08685140705, 46293560.87307053]
Total newton steps:  25
***Optimal point of x1 and x2 =  [-0.48672345  0.26348271]
***Optimal of w = x3 =  [5.78669148]
***Optimal value =  [5.78669148]
=================================================
### For u = 2000:
Iters for outer loop:  3
Inner loops for each outer loop:  [5, 9, 8]
t:  [1.0, 2000.0, 4000000.0]
f(x):  [4.541960777586963, 11587.82834071653, 23146794.56739345]
Total newton steps:  22
***Optimal point of x1 and x2 =  [-0.4867234   0.26348272]
***Optimal of w = x3 =  [5.78669173]
***Optimal value =  [5.78669173]
=================================================
### For u = 20000:
Iters for outer loop:  2
Inner loops for each outer loop:  [5, 12]
t:  [1.0, 20000.0]
f(x):  [4.541960777586963, 115752.87593957357]
Total newton steps:  17
***Optimal point of x1 and x2 =  [-0.48670416  0.26348488]
***Optimal of w = x3 =  [5.78679122]
***Optimal value =  [5.78679122]
=================================================
### Using cvx tool...
***Optimal point of x1 and x2 =  [-0.48678097  0.26347138]
***Optimal of w = x3 =  5.786691209246071
***Optimal value =  5.786691209246071
```

(n) 到小數點後 3~4 位都差異不大 (for optimal value and optimal point)

μ	20	200	2000	20000
optimal value	5.78669126	5.78669148	5.78669173	5.78679122
optimal point	[-0.48672349, 0.26348271]	[-0.48672345, 0.26348271]	[-0.4867234, 0.26348272]	[-0.48670416, 0.26348488]

(o) $\mu = 20000$ used the lowest total number of Newton steps.

(p) cvx_method () in hw3_main.py

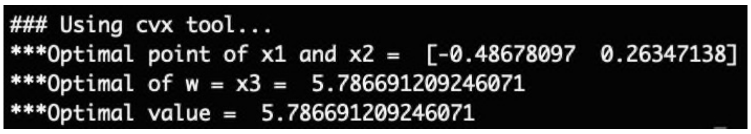
⇒ very similar to the optimal point and optimal value obtained by the log barrier method.

$\boxed{\text{Barrier Method}}$

μ	20	200	2000	20000
optimal value	5.78669126	5.78669148	5.78669173	5.78679122
optimal point	[-0.48672349, 0.26348271]	[-0.48672345, 0.26348271]	[-0.4867234, 0.26348272]	[-0.48670416, 0.26348488]

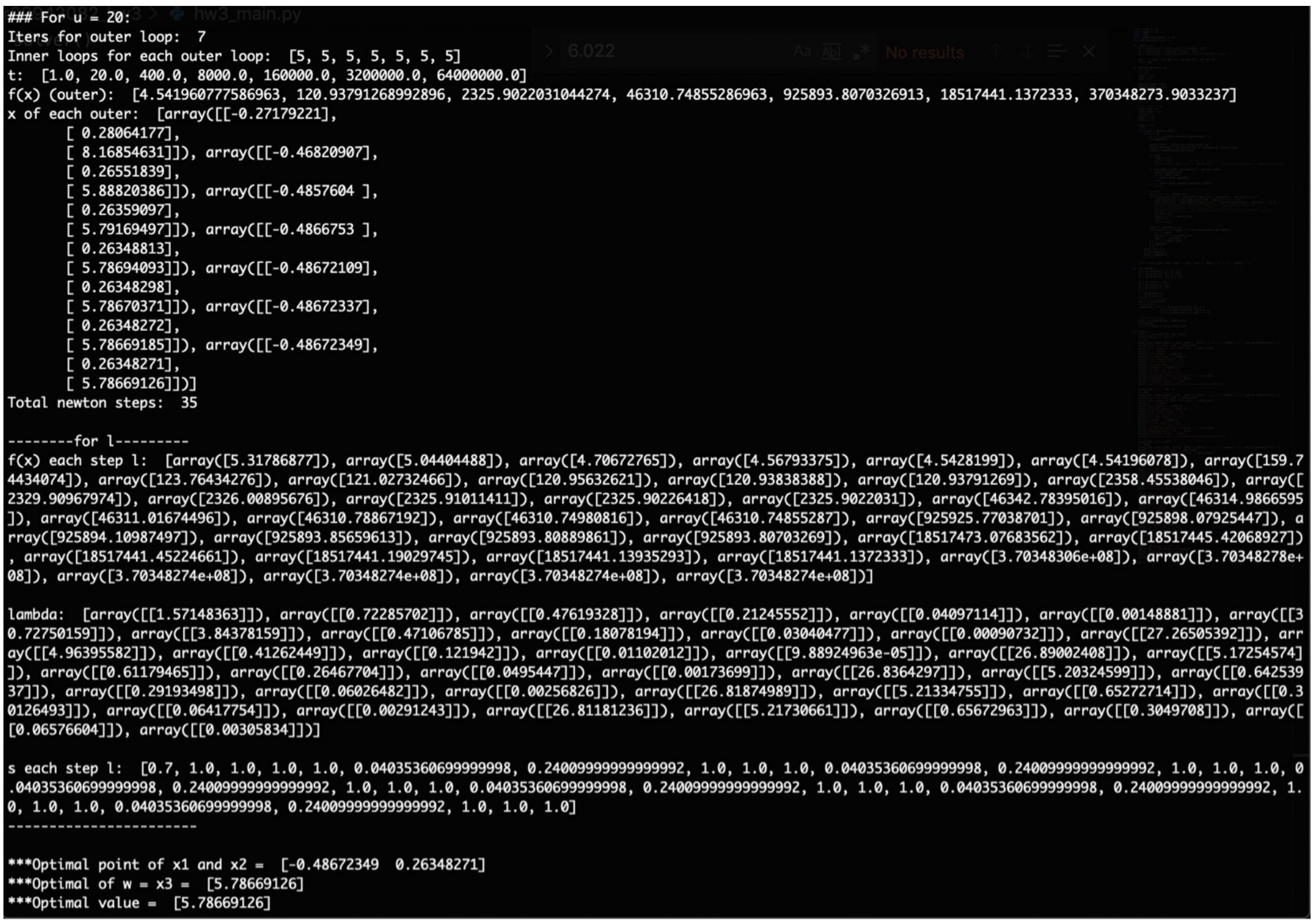
$\boxed{\text{CVX tool}}$

optimal value	5.78669120924607)
optimal point	[-0.48678097, 0.26347138]



```
### Using cvx tool...
***Optimal point of x1 and x2 =  [-0.48678097  0.26347138]
***Optimal of w = x3 =  5.786691209246071
***Optimal value =  5.786691209246071
```

$\boxed{\text{完整 log of code}}$ include $f(x^{(l)})$, $\lambda(x^{(l)})$, $s^{(l)}$



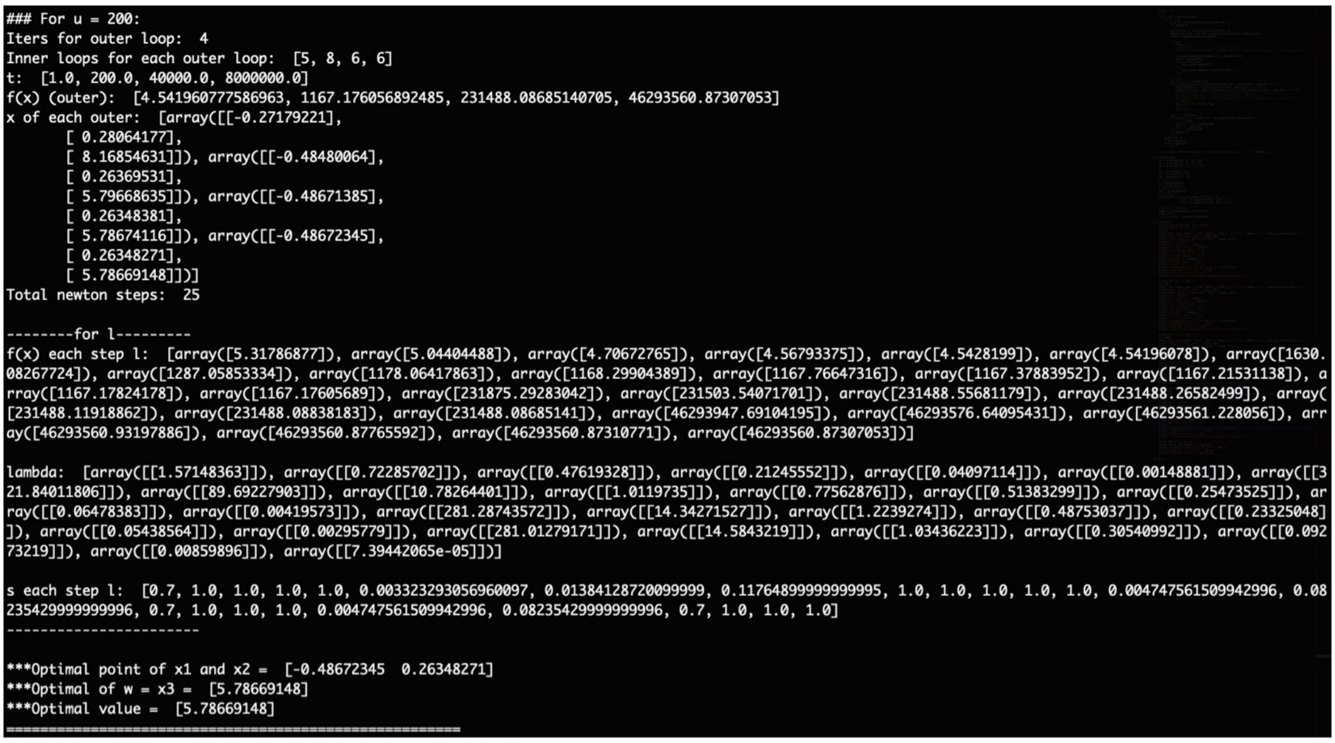
```
### For u = 20:
Iters for outer loop:  7
Inner loops for each outer loop:  [5, 5, 5, 5, 5, 5, 5]
t:  [1.0, 20.0, 400.0, 8000.0, 160000.0, 3200000.0, 64000000.0]
f(x) (outer):  [4.541960777586963, 120.93791268992896, 2325.9022031044274, 46310.74855286963, 925893.8070326913, 18517441.1372333, 370348273.9033237]
x of each outer:  [array([[-0.27179221],
       [ 0.28064177],
       [ 8.16854631]]), array([[-0.46820907],
       [ 0.26551839],
       [ 5.88820386]]), array([[-0.4857604 ],
       [ 0.26359097],
       [ 5.79169497]]), array([[-0.4866753 ],
       [ 0.26348813],
       [ 5.78694093]]), array([[-0.48672109],
       [ 0.26348298],
       [ 5.78670371]]), array([[-0.48672337],
       [ 0.26348272],
       [ 5.78669185]]), array([[-0.48672349],
       [ 0.26348271],
       [ 5.78669126]])]
Total newton steps:  35

--------for l---------
f(x) each step l:  [array([5.31786877]), array([5.04404488]), array([4.70672765]), array([4.56793375]), array([4.5428199]), array([4.54196078]), array([159.7
4434074]), array([123.76434276]), array([121.02732466]), array([120.95632621]), array([120.93838388]), array([120.93791269]), array([2358.45538046]), array([
2329.90967974]), array([2326.00895676]), array([2325.91011411]), array([2325.90226418]), array([2325.9022031]), array([46342.78395016]), array([46314.9866595
]), array([46311.01674496]), array([46310.78867192]), array([46310.74980816]), array([46310.74855287]), array([925925.77038701]), array([925898.07925447]), a
rray([925894.10987497]), array([925893.85659613]), array([925893.80889861]), array([925893.80703269]), array([18517473.07683562]), array([18517445.42068927])
, array([18517441.45224661]), array([18517441.19029745]), array([18517441.13935293]), array([18517441.1372333]), array([3.70348306e+08]), array([3.70348278e+
08]), array([3.70348274e+08]), array([3.70348274e+08]), array([3.70348274e+08]), array([3.70348274e+08])]

lambda:  [array([[1.57148363]]), array([[0.72285702]]), array([[0.47619328]]), array([[0.21245552]]), array([[0.04097114]]), array([[0.00148881]]), array([[3
0.72750159]]), array([[3.84378159]]), array([[0.47106785]]), array([[0.18078194]]), array([[0.03040477]]), array([[0.00090732]]), array([[27.26505392]]), arr
ay([[4.96395582]]), array([[0.41262449]]), array([[0.121942]]), array([[0.01102012]]), array([[9.88924963e-05]]), array([[26.89002408]]), array([[5.17254574]
]), array([[0.61179465]]), array([[0.26467704]]), array([[0.0495447]]), array([[0.00173699]]), array([[26.8364297]]), array([[5.20324599]]), array([[0.642539
37]]), array([[0.29193498]]), array([[0.06026482]]), array([[0.00256826]]), array([[26.81874989]]), array([[5.21334755]]), array([[0.65272714]]), array([[0.3
0126493]]), array([[0.06417754]]), array([[0.00291243]]), array([[26.81181236]]), array([[5.21730661]]), array([[0.65672963]]), array([[0.3049708]]), array([
[0.06576604]]), array([[0.00305834]])]

s each step l:  [0.7, 1.0, 1.0, 1.0, 1.0, 0.04035360699999998, 0.24009999999999992, 1.0, 1.0, 1.0, 0.04035360699999998, 0.24009999999999992, 1.0, 1.0, 1.0, 0
.04035360699999998, 0.24009999999999992, 1.0, 1.0, 1.0, 0.04035360699999998, 0.24009999999999992, 1.0, 1.0, 1.0, 0.04035360699999998, 0.24009999999999992, 1.
0, 1.0, 1.0, 0.04035360699999998, 0.24009999999999992, 1.0, 1.0, 1.0]
-----------------------

***Optimal point of x1 and x2 =  [-0.48672349  0.26348271]
***Optimal of w = x3 =  [5.78669126]
***Optimal value =  [5.78669126]
```



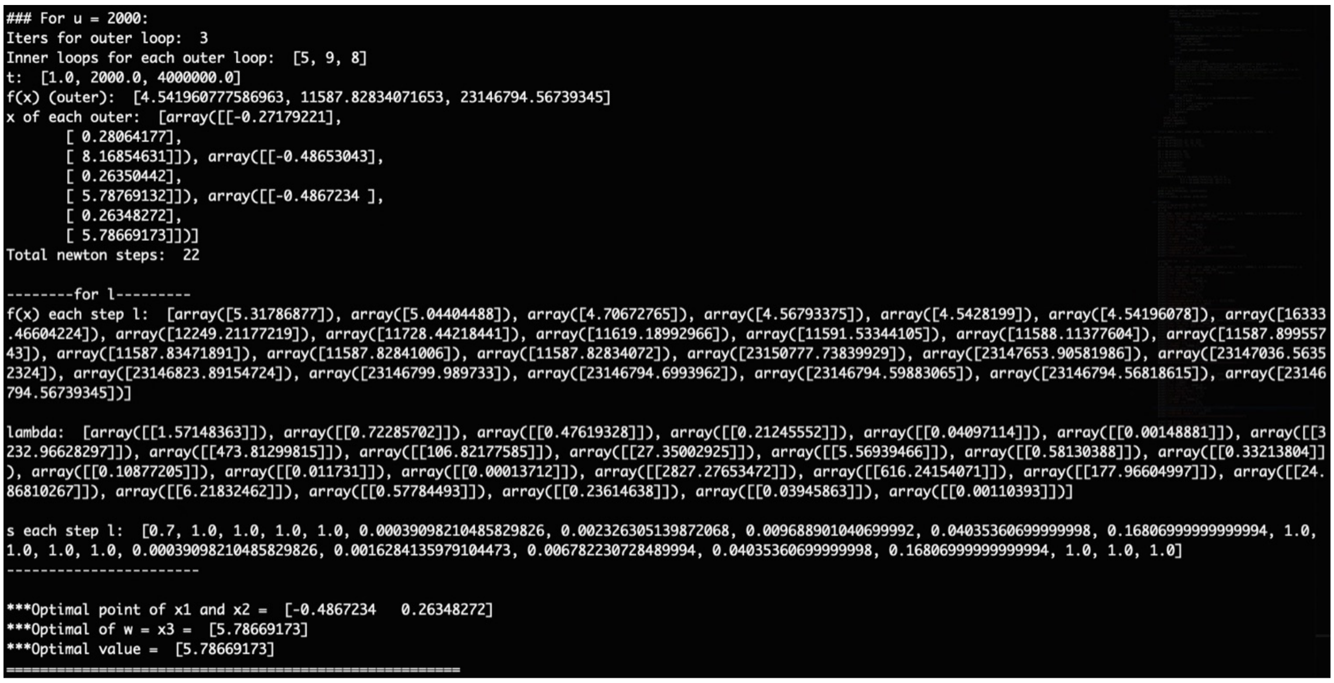
```
### For u = 200:
Iters for outer loop:  4
Inner loops for each outer loop:  [5, 8, 6, 6]
t:  [1.0, 200.0, 40000.0, 8000000.0]
f(x) (outer):  [4.541960777586963, 1167.176056892485, 231488.08685140705, 46293560.87307053]
x of each outer:  [array([[-0.27179221],
       [ 0.28064177],
       [ 8.16854631]]), array([[-0.48480064],
       [ 0.26369531],
       [ 5.79668635]]), array([[-0.48671385],
       [ 0.26348381],
       [ 5.78674116]]), array([[-0.48672345],
       [ 0.26348271],
       [ 5.78669148]])]
Total newton steps:  25

--------for l---------
f(x) each step l:  [array([5.31786877]), array([5.04404488]), array([4.70672765]), array([4.56793375]), array([4.5428199]), array([4.54196078]), array([1630.
08267724]), array([1287.05853334]), array([1178.06417863]), array([1168.29904389]), array([1167.76647316]), array([1167.37883952]), array([1167.21531138]), a
rray([1167.17824178]), array([1167.17605689]), array([231875.29283042]), array([231503.54071701]), array([231488.55681179]), array([231488.26582499]), array(
[231488.11918862]), array([231488.08838183]), array([231488.08685141]), array([46293947.69104195]), array([46293576.64095431]), array([46293561.228056]), arr
ay([46293560.93197886]), array([46293560.87765592]), array([46293560.87310771]), array([46293560.87307053])]

lambda:  [array([[1.57148363]]), array([[0.72285702]]), array([[0.47619328]]), array([[0.21245552]]), array([[0.04097114]]), array([[0.00148881]]), array([[3
21.84011806]]), array([[89.69227903]]), array([[10.78264401]]), array([[1.0119735]]), array([[0.77562876]]), array([[0.51383299]]), array([[0.25473525]]), ar
ray([[0.06478383]]), array([[0.00419573]]), array([[281.28743572]]), array([[14.34271527]]), array([[1.2239274]]), array([[0.48753037]]), array([[0.23325048]
]), array([[0.05438564]]), array([[0.00295779]]), array([[281.01279171]]), array([[14.5843219]]), array([[1.03436223]]), array([[0.30540992]]), array([[0.092
73219]]), array([[0.00859896]]), array([[7.39442065e-05]])]

s each step l:  [0.7, 1.0, 1.0, 1.0, 1.0, 0.003323293056960097, 0.013841287200999999, 0.11764899999999995, 1.0, 1.0, 1.0, 1.0, 1.0, 0.004747561509942996, 0.08
235429999999996, 0.7, 1.0, 1.0, 1.0, 0.004747561509942996, 0.08235429999999996, 0.7, 1.0, 1.0, 1.0]
-----------------------

***Optimal point of x1 and x2 =  [-0.48672345  0.26348271]
***Optimal of w = x3 =  [5.78669148]
***Optimal value =  [5.78669148]
==================================================
```



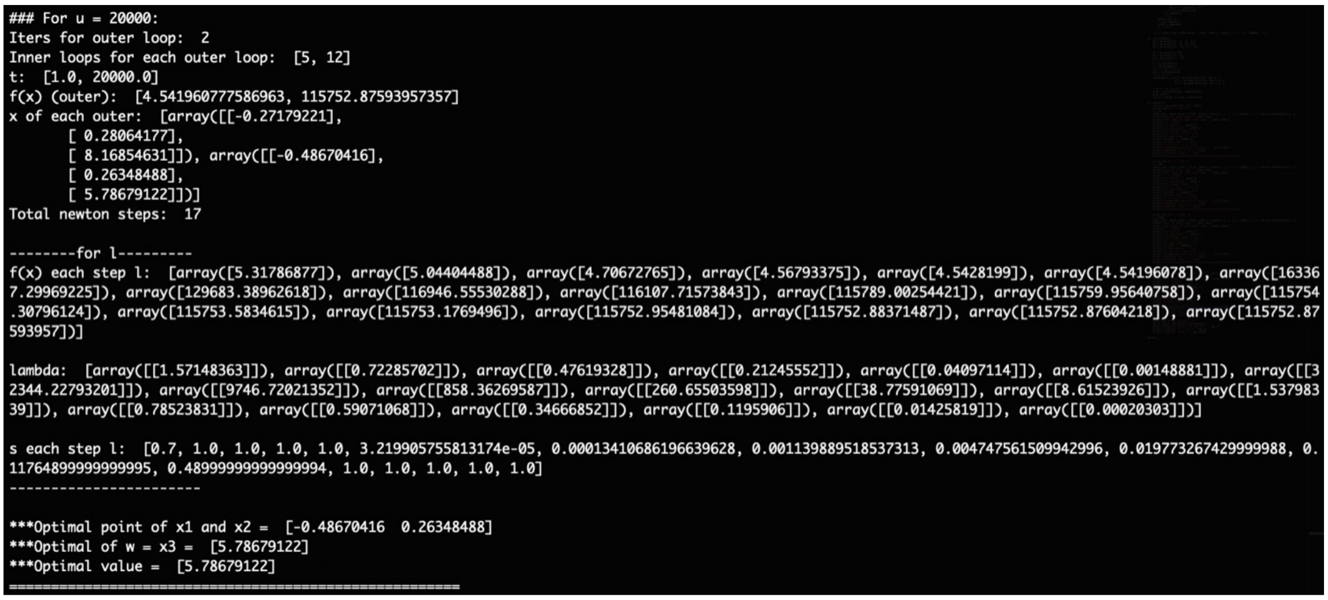
```
### For u = 2000:
Iters for outer loop:  3
Inner loops for each outer loop:  [5, 9, 8]
t:  [1.0, 2000.0, 4000000.0]
f(x) (outer):  [4.541960777586963, 11587.82834071653, 23146794.56739345]
x of each outer:  [array([[-0.27179221],
       [ 0.28064177],
       [ 8.16854631]]), array([[-0.48653043],
       [ 0.26350442],
       [ 5.78769132]]), array([[-0.4867234 ],
       [ 0.26348272],
       [ 5.78669173]])]
Total newton steps:  22

--------for l---------
f(x) each step l:  [array([5.31786877]), array([5.04404488]), array([4.70672765]), array([4.56793375]), array([4.5428199]), array([4.54196078]), array([16333
.46604224]), array([12249.21177219]), array([11728.44218441]), array([11619.18992966]), array([11591.53344105]), array([11588.11377604]), array([11587.899557
43]), array([11587.83471891]), array([11587.82841006]), array([11587.82834072]), array([23150777.73839929]), array([23147653.90581986]), array([23147036.5635
2324]), array([23146823.89154724]), array([23146799.989733]), array([23146794.6993962]), array([23146794.59883065]), array([23146794.56818615]), array([23146
794.56739345])]

lambda:  [array([[1.57148363]]), array([[0.72285702]]), array([[0.47619328]]), array([[0.21245552]]), array([[0.04097114]]), array([[0.00148881]]), array([[3
232.96628297]]), array([[473.81299815]]), array([[106.82177585]]), array([[27.35002925]]), array([[5.56939466]]), array([[0.58130388]]), array([[0.33213804]]
), array([[0.10877205]]), array([[0.011731]]), array([[0.00013712]]), array([[2827.27653472]]), array([[616.24154071]]), array([[177.96604997]]), array([[24.
86810267]]), array([[6.21832462]]), array([[0.57784493]]), array([[0.23614638]]), array([[0.03945863]]), array([[0.00110393]])]

s each step l:  [0.7, 1.0, 1.0, 1.0, 1.0, 0.00039098210485829826, 0.002326305139872068, 0.009688901040699992, 0.04035360699999998, 0.16806999999999994, 1.0,
1.0, 1.0, 1.0, 0.00039098210485829826, 0.0016284135979104473, 0.006782230728489994, 0.04035360699999998, 0.16806999999999994, 1.0, 1.0, 1.0]
-----------------------

***Optimal point of x1 and x2 =  [-0.4867234   0.26348272]
***Optimal of w = x3 =  [5.78669173]
***Optimal value =  [5.78669173]
==================================================
```



```
### For u = 20000:
Iters for outer loop:  2
Inner loops for each outer loop:  [5, 12]
t:  [1.0, 20000.0]
f(x) (outer):  [4.541960777586963, 115752.87593957357]
x of each outer:  [array([[-0.27179221],
       [ 0.28064177],
       [ 8.16854631]]), array([[-0.48670416],
       [ 0.26348488],
       [ 5.78679122]])]
Total newton steps:  17

--------for l---------
f(x) each step l:  [array([5.31786877]), array([5.04404488]), array([4.70672765]), array([4.56793375]), array([4.5428199]), array([4.54196078]), array([16336
7.29969225]), array([129683.38962618]), array([116946.55530288]), array([116107.71573843]), array([115789.00254421]), array([115759.95640758]), array([115754
.30796124]), array([115753.5834615]), array([115753.1769496]), array([115752.95481084]), array([115752.88371487]), array([115752.87604218]), array([115752.87
593957])]

lambda:  [array([[1.57148363]]), array([[0.72285702]]), array([[0.47619328]]), array([[0.21245552]]), array([[0.04097114]]), array([[0.00148881]]), array([[3
2344.22793201]]), array([[9746.72021352]]), array([[858.36269587]]), array([[260.65503598]]), array([[38.77591069]]), array([[8.61523926]]), array([[1.537983
39]]), array([[0.78523831]]), array([[0.59071068]]), array([[0.34666852]]), array([[0.1195906]]), array([[0.01425819]]), array([[0.00020303]])]

s each step l:  [0.7, 1.0, 1.0, 1.0, 1.0, 3.219905755813174e-05, 0.00013410686196639628, 0.0011398895185373l3, 0.004747561509942996, 0.019773267429999988, 0.
11764899999999995, 0.48999999999999994, 1.0, 1.0, 1.0, 1.0, 1.0]
-----------------------

***Optimal point of x1 and x2 =  [-0.48670416  0.26348488]
***Optimal of w = x3 =  [5.78679122]
***Optimal value =  [5.78679122]
==================================================
```