ECE 647: Programming Project

Spring 2025

Submitted to: Prof. Christopher Brinton

Submitted by: Rittwik Sood

Email: sood26@purdue.edu

PUID: 036403763

Link: https://github.com/rittwiksood/Convex_Optimization_Prog_Project.git

Index…………………………………………………………………………………….

1. Problem 1 : https://github.com/rittwiksood/Convex_Optimization_Prog_Project/blob/main/RittwikSood_Q1.m
2. Problem 2: https://github.com/rittwiksood/Convex_Optimization_Prog_Project/blob/main/RittwikSood_Q2.m
3. Results: https://github.com/rittwiksood/Convex_Optimization_Prog_Project/tree/main/images

………………………………………………………………………………………….

Readme:

On the git project, This project consists of:

a.) Given files
b.) RittwikSood_Q1.m : Question 1 code
c.) RittwikSood_Q2.m : Question 2 code
d.) Dependent files: Added function files in separate .m files
e.) Images folder: Contains all the results (All the plots and figures provided in this report)

Problem 1

a) $f(x_1, x_2) = x_1^2 + 3x_1x_2 + 9x_2^2 + 2x_1 - 5x_2$

Using mesh() function to plot Figure 1 below. We can see the function is convex. (**LINK** to the figures)

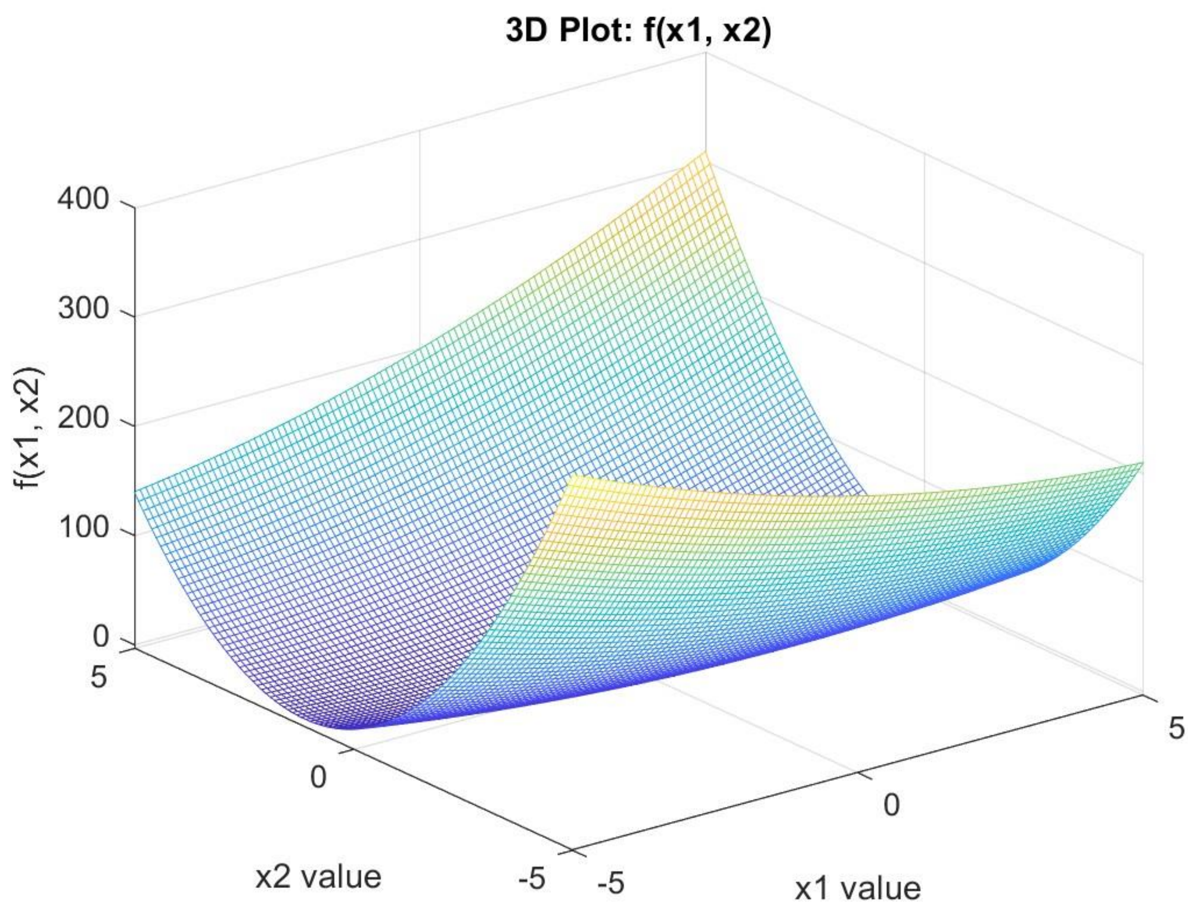

Figure 1

Directions used here: DIRECTIONS = [1, 1; -1, 1; 2, -1];

i.) For direction [1,1]:

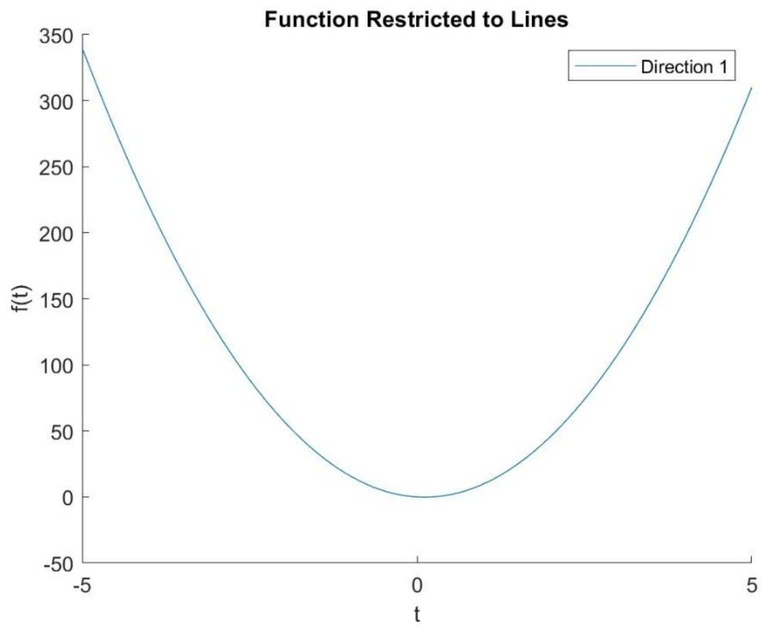

Figure 2

ii.) For direction [-1,1]:

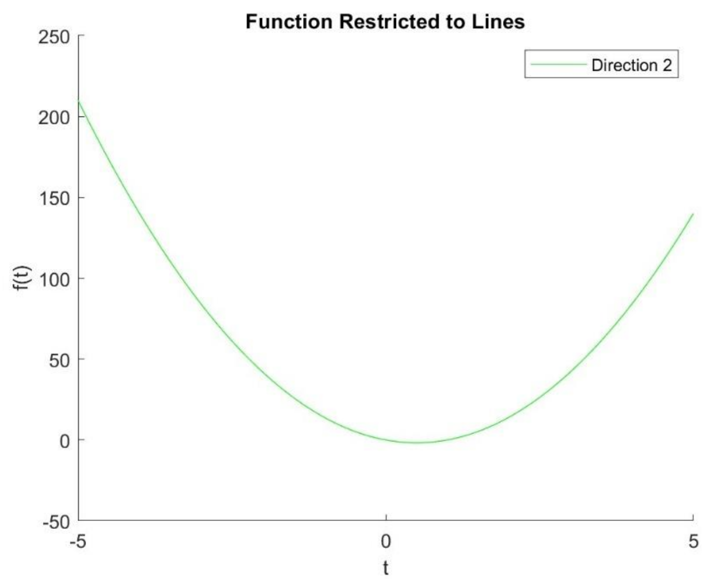

Figure 3

iii.) For direction [2, -1]:

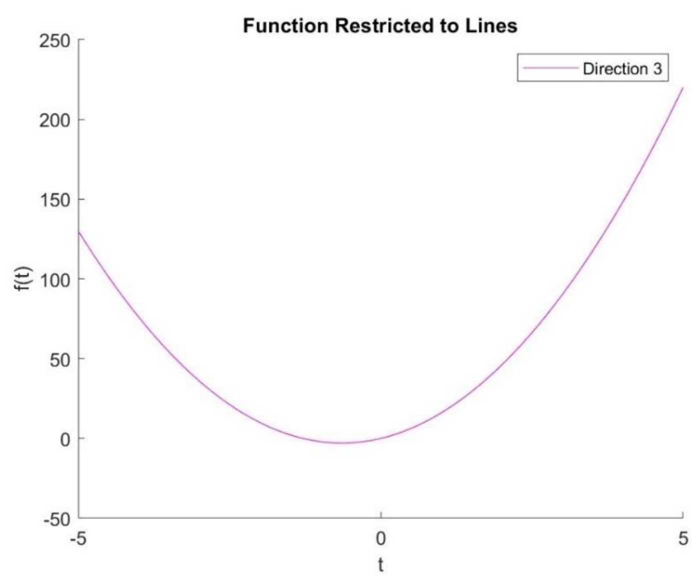

Figure 4

We can conclude from the above figures (and below one, all concatenated) that the function in any direction is **convex**.

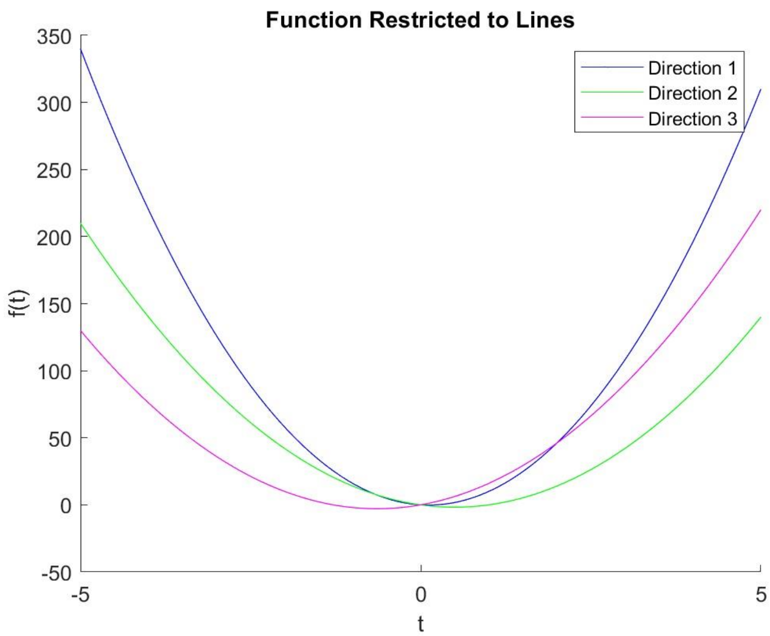

Figure 5

Problem 1 b)

- **Too small** step size is considered as 0.001, in which case, even after 1000 iterations, it does not converge to the optimal point, but goes towards it.

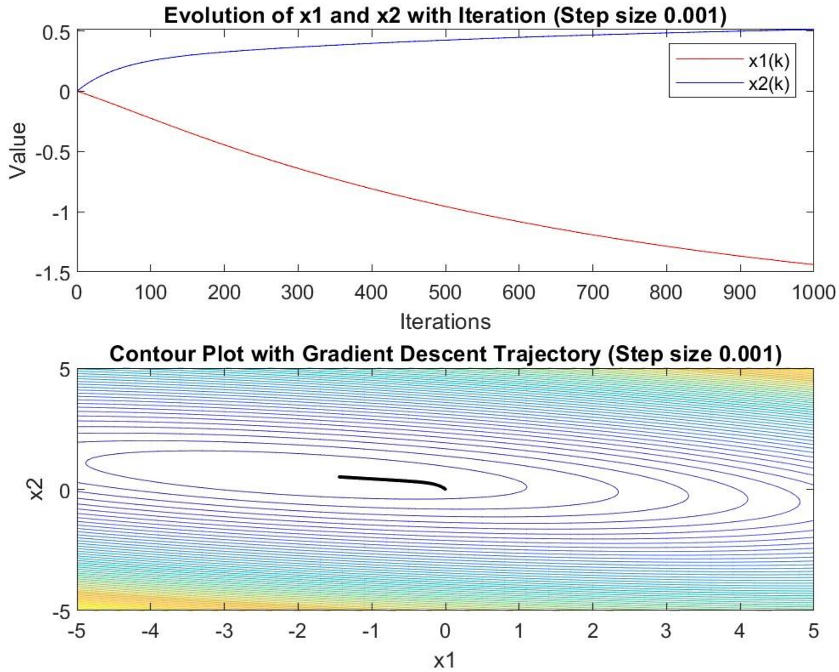

Figure 6

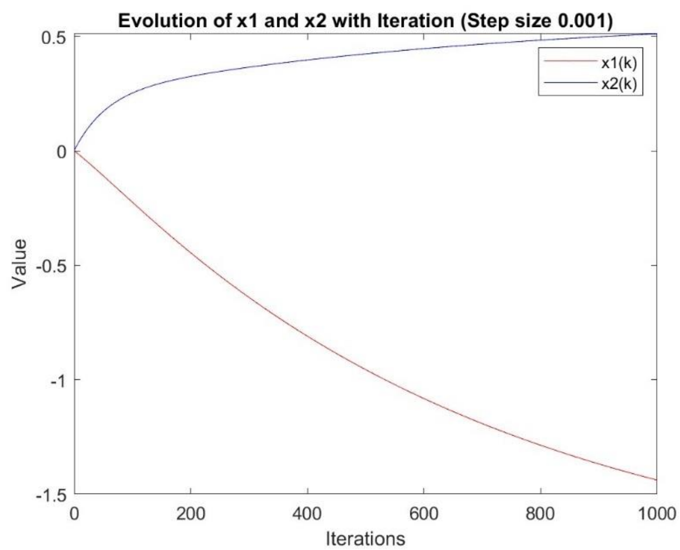

Figure 7

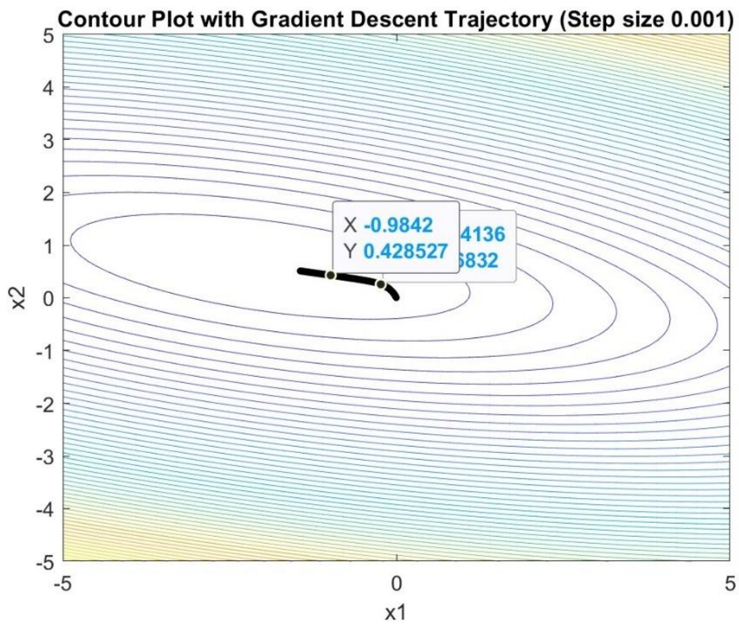

Figure 8

- **Too large** step size is considered as 0.11, in which case both x1 and x2 diverges.

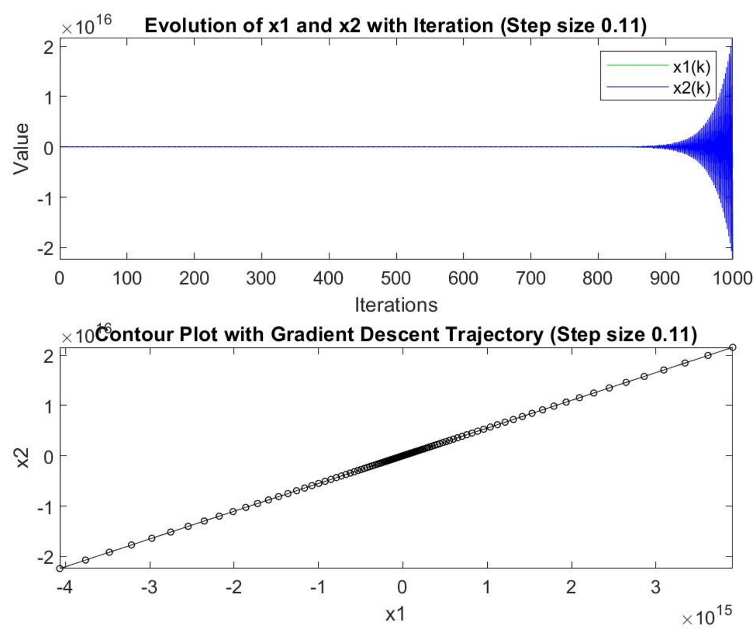

Figure 9

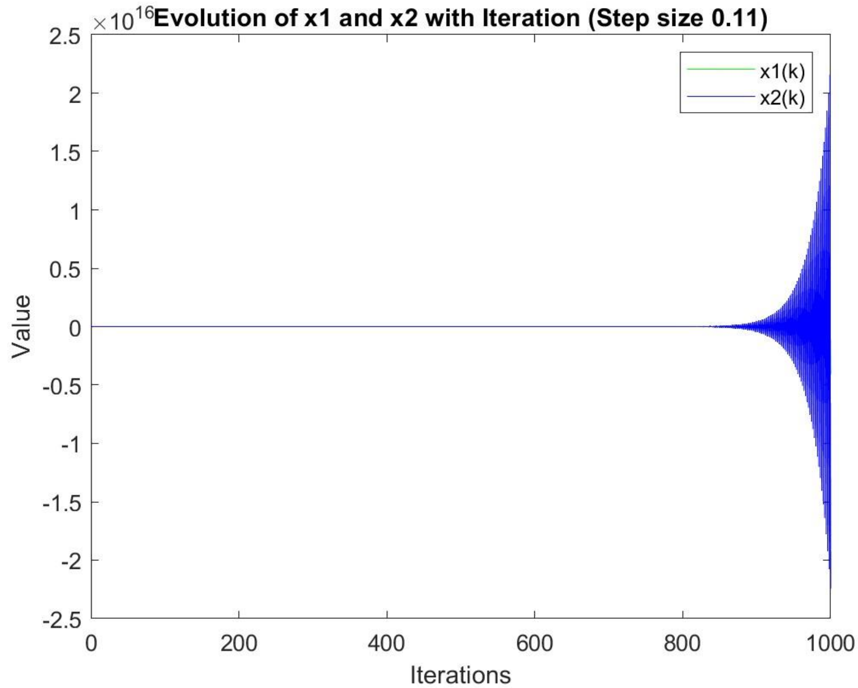

Figure 10

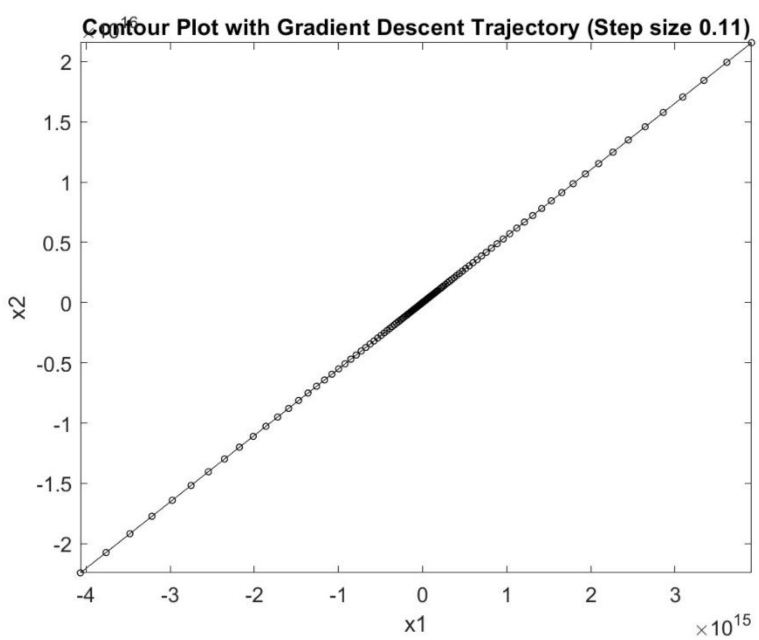

Figure 11

- Perfect step size is 0.1 . Which converges to optimal value: [-1.8824, 0.5925]

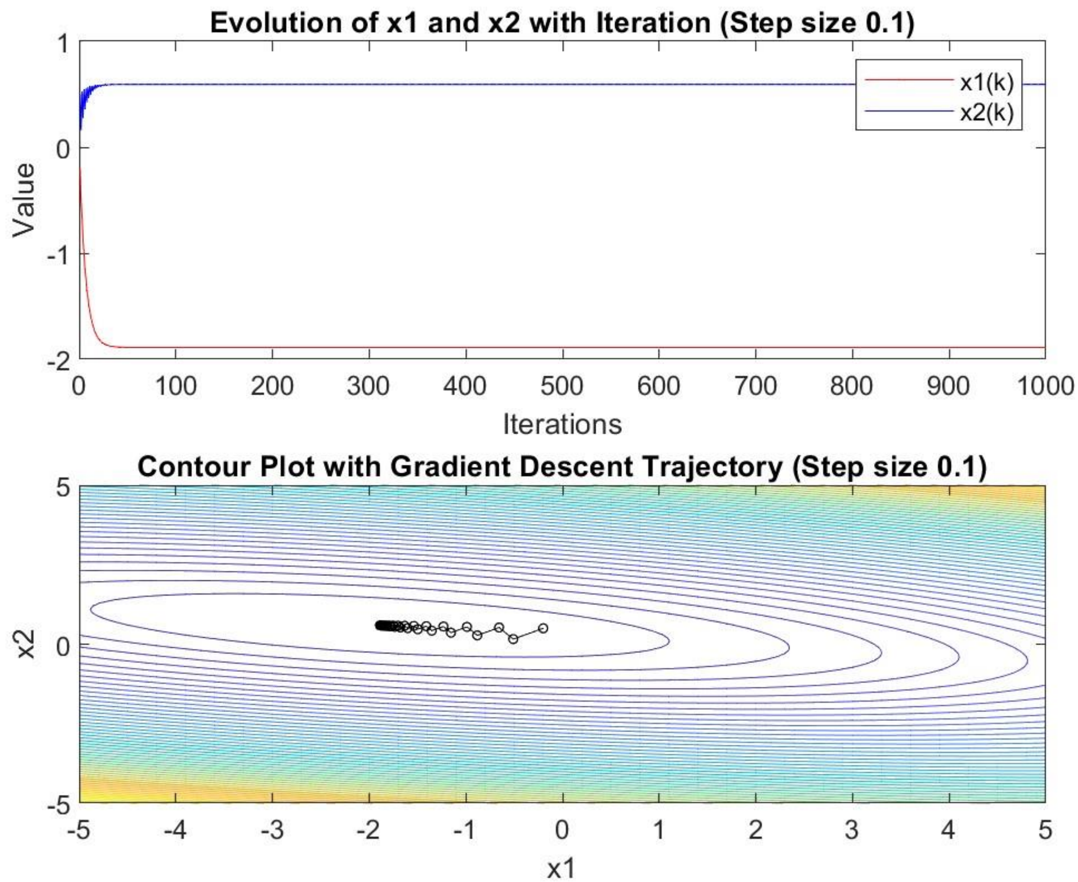

Figure 12

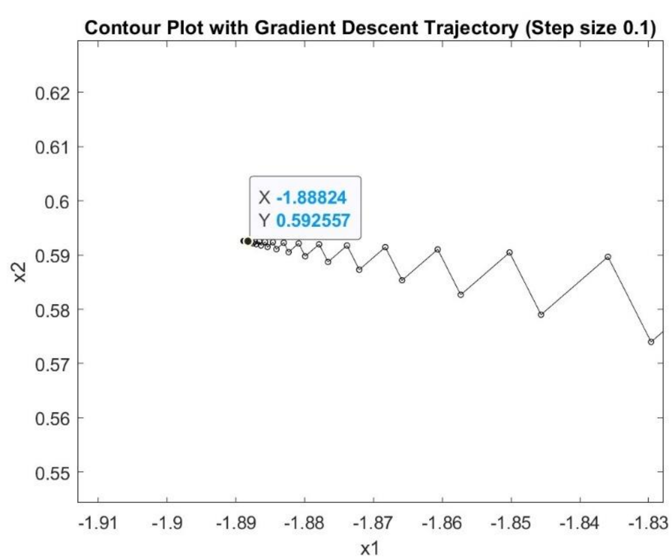

Figure 13

Problem 1 c)

- Gamma values taken = { 0.001, 0.1, 10.0}
- Small step size = 0.001
- Perfect step size = 0.1
- Large step size = 10

For Small step size, 0.001

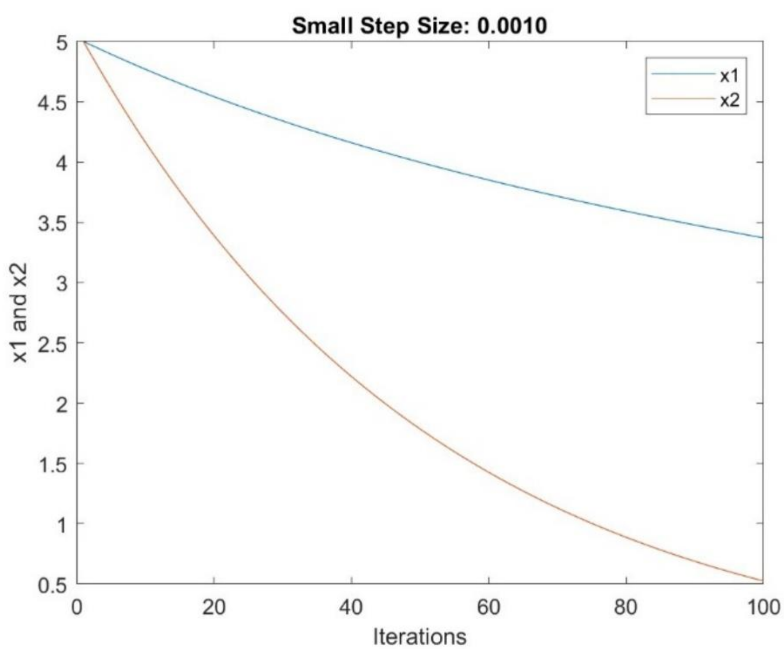

Figure 14

For Large step size = 10

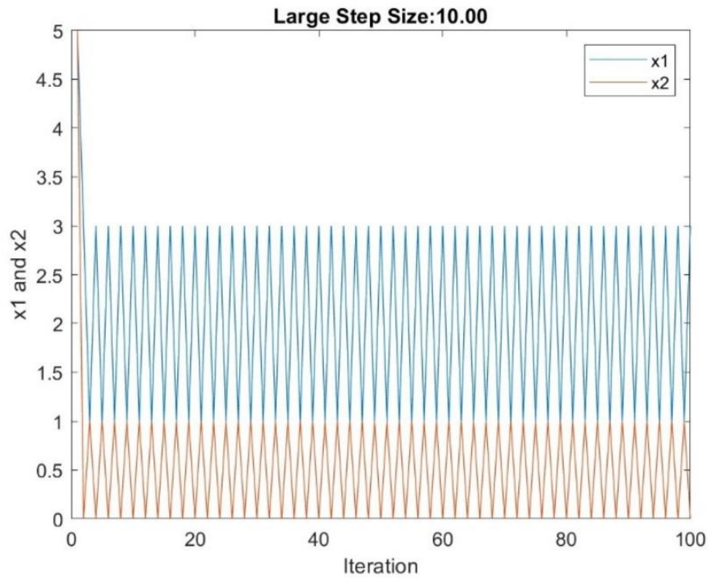

Figure 15

Perfect step size = 0.1

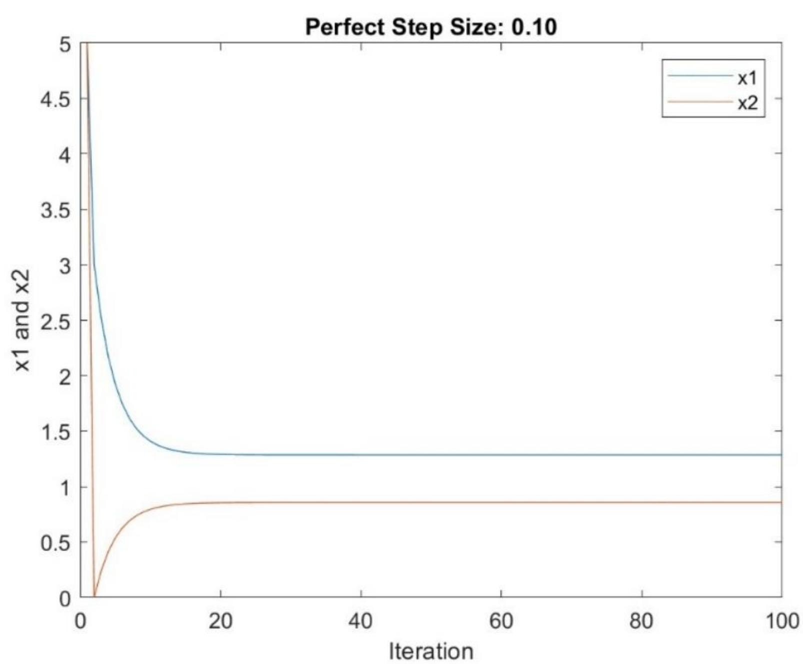

Figure 16

Optimal solution comes out to be for perfect gamma value = [1.285714, 0.857143]. Figure 17 shows the optimum value and Figure 18 shows the contour plots of different step size values.

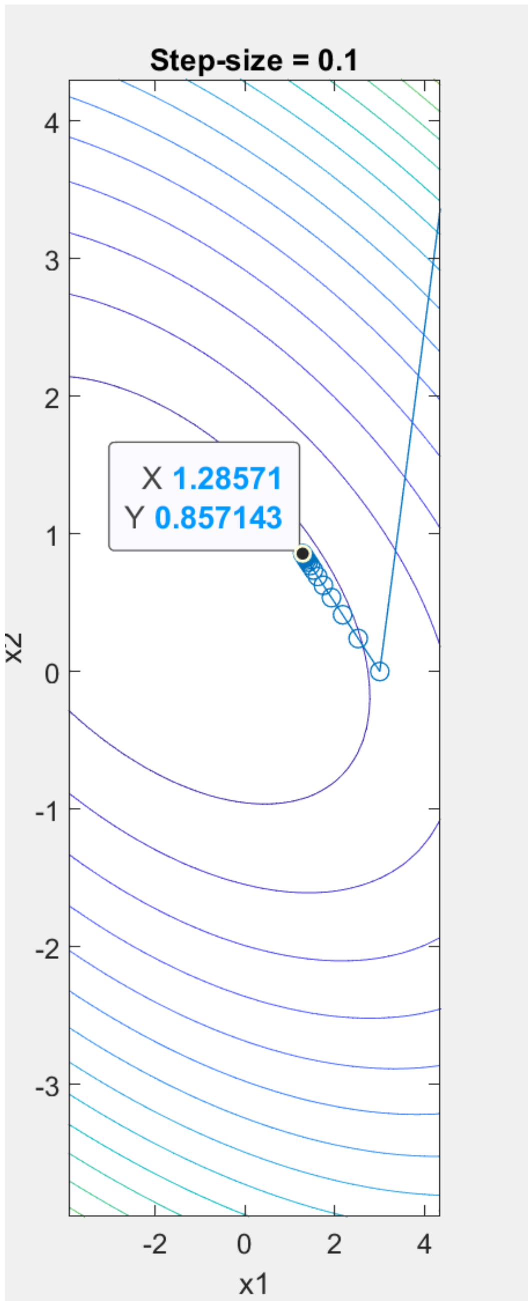

Figure 17

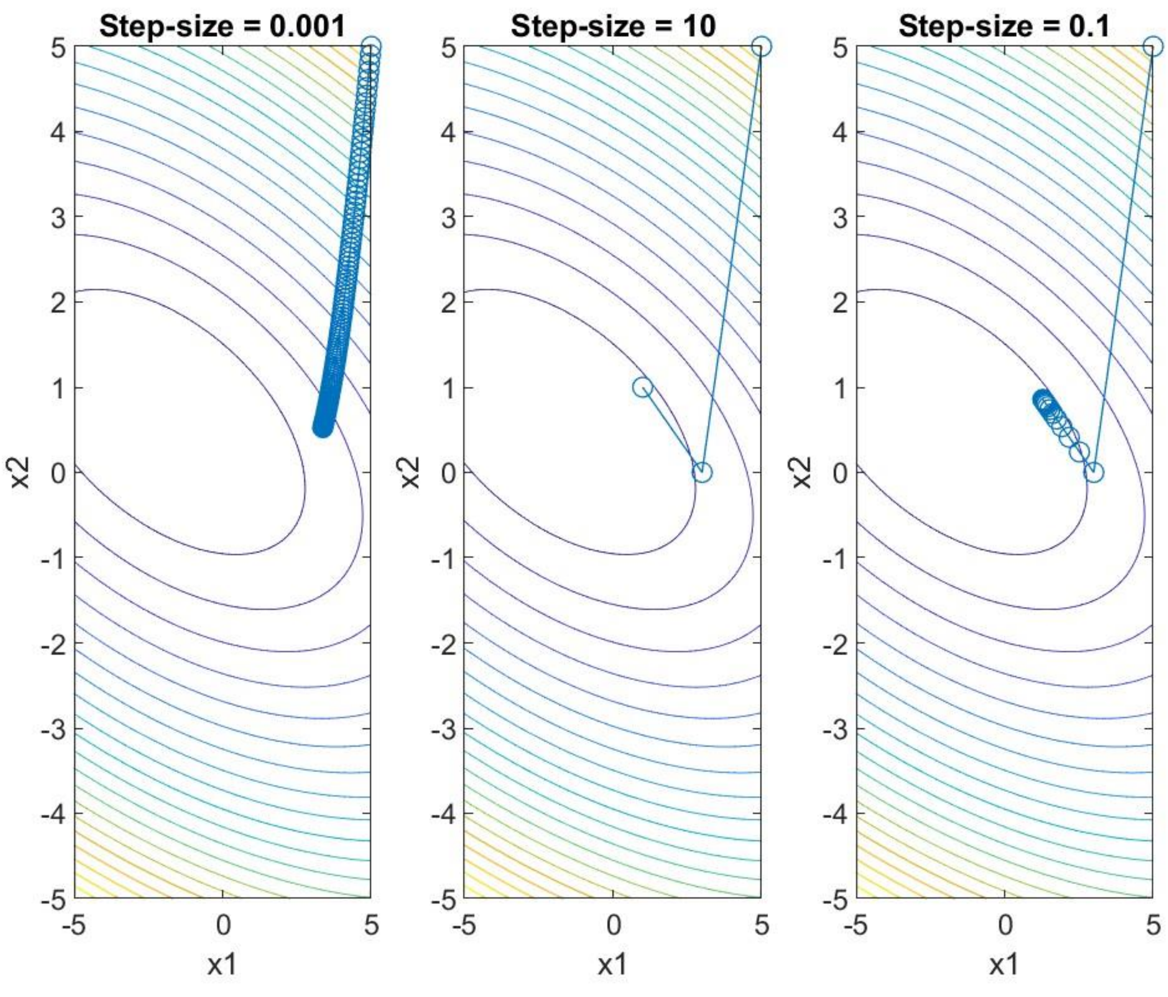

Figure 18

Problem 1 d)

- Gamma values taken = { 0.005, 0.1, 1.5}
- Small step size = 0.005
- Perfect step size = 0.1
- Large step size = 1.5

I.) Trajectory of Primal variables for the above gamma values across iterations is given by Figure 19.

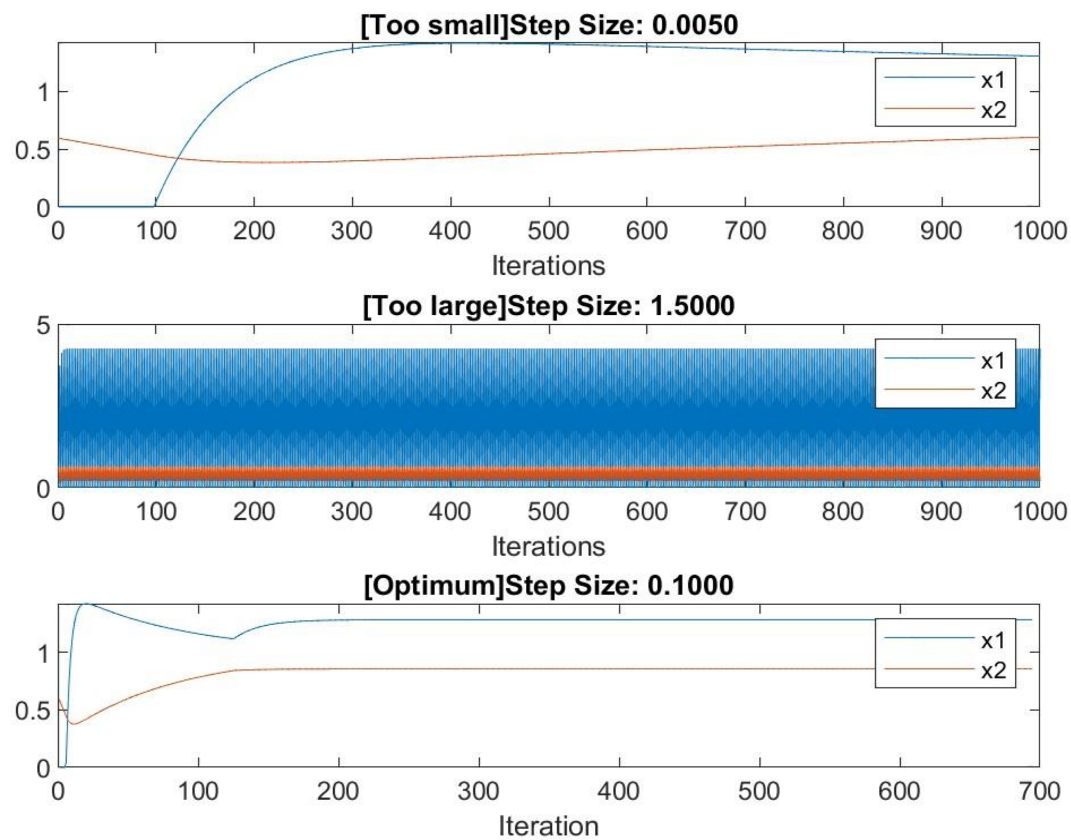

Figure 19

II.) Figure 20 represents dual variables change with the number of iterations.

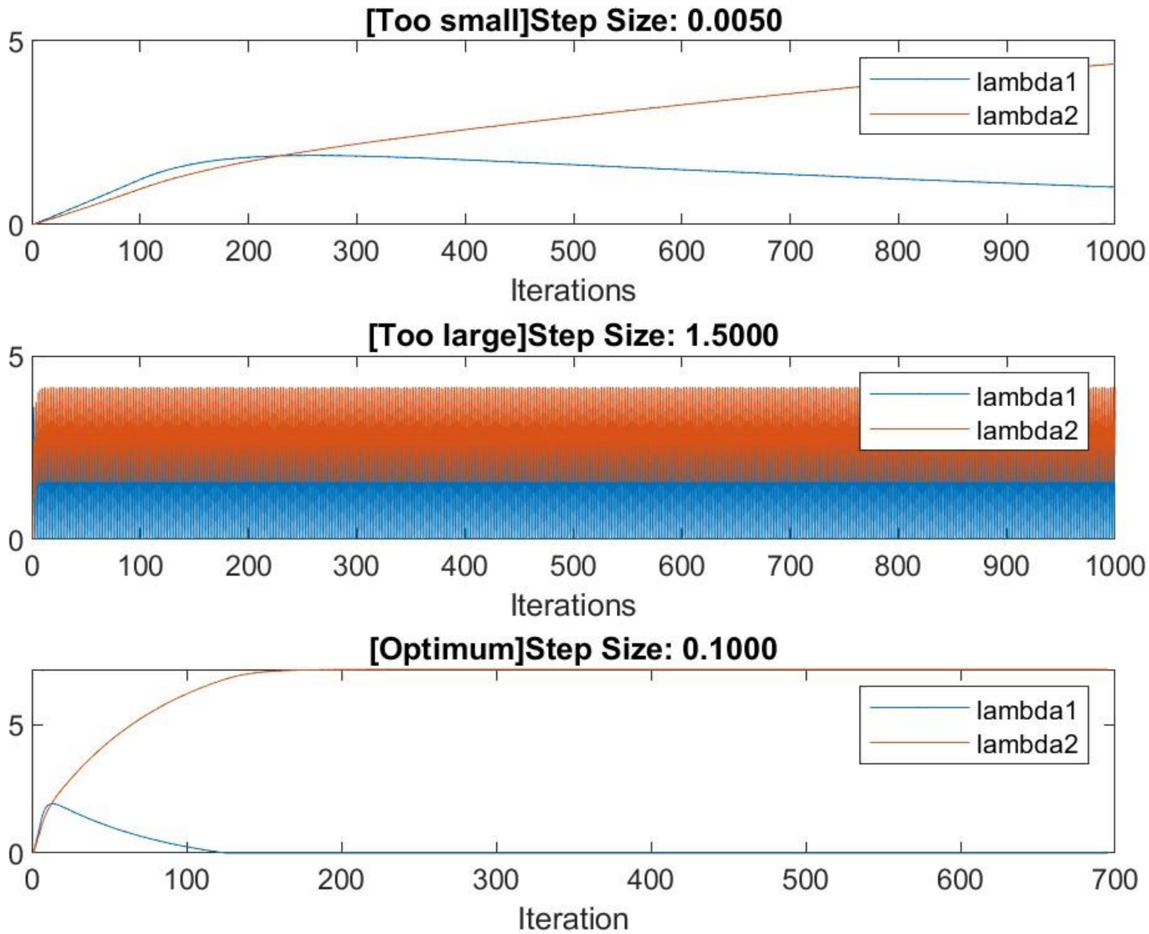

Figure 20

III.) Figure 21 represents all the values of primal and dual variables being plotted together for three different gammas.

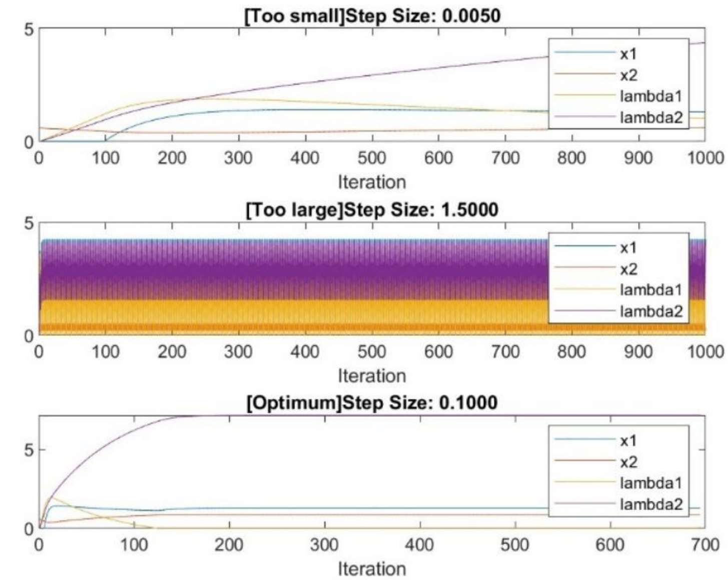

Figure 21

IV.) Figure 22 represents the plot the trajectory of (x1; x2) on the contour plot too. (For three different set of gammas)

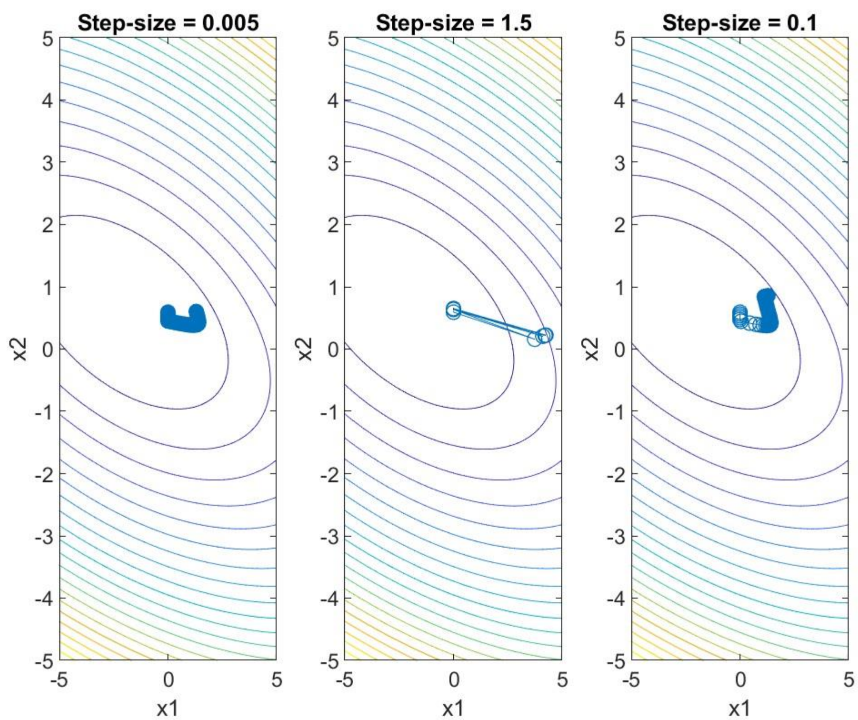

Figure 22

Problem 1 e)

Optimal solution: [1.285714, 0.857143]

Objective value: 9.857143

Optimal solution: [1.285714, 0.857143]

Optimal dual variable: [0.000000, 7.142857]

Objective value: 9.857143

check Part 1.b

Optimal solution is : [-1.887802, 0.592395]

Objective value is : -3.370369

Gradient: [0.001583, -0.000287]

Gradient of function is zero: true

check Part 1.c

Optimal solution: [1.285714, 0.857143]

Objective value: 9.857143

Constraint satisfied for constraint line: [-2, 1]

Gradient: [7.142857, 14.285715]

Inner product of gradient and above constraint line is zero: true

check Part 1.d

Optimal solution: [1.285714, 0.857143]

Optimal dual variable: [0.000000, 7.142857]

Objective value: 9.857143

Primal feasibility: true

Dual feasibility: true

Complementary slackness: true

Gradient of Lagrangian function: [0.000000, -0.000000]

Gradient of Lagrangian function is zero: true

Solution is optimal: true

Solution is optimal: Problem 1 ends. Thanks!!>>

Problem 2

a) Formulate the rate control problem as a utility maximization problem.

We are to maximize total utility across 7 flows, where utility function for flow i is:

$$U_i(x_i) = w_i \ln (x_i) \qquad - (1)$$

Subject to link capacity constraints.

Let:

- x_i : *flow rate for flow i (scalar)*
- w_i : *weight for flow i*
- $R \in \mathbb{R}^{12\times 7}$,routing matrix, where R_{ji} = 1 if flow i uses link *j*
- $c \in \mathbb{R}^{12}$, vector of link capacities
- $x \in \mathbb{R}^{7}$, flow rate vector

Primal Problem (Utility Maximization)

The objective function thus becomes:

$$\max_{x>o} \sum_{i=1}^{7} w_i \, ln(x_i) \qquad - (2)$$

subject to $Rx \leq c$

i.e.

a.) $x_i \geq 0$ for all *i* (flow rates cannot be negative)

b.) $Rx \leq c$ for all *j* (the total flow rate on each link cannot exceed its capacity C_j

Now, we'll derive the dual gradient algorithm to solve this problem. First, we introduce the Lagrange multipliers (λ) for the link capacity constraints. The Lagrangian for the problem is:

$$\mathcal{L}(x,\lambda) = \sum_{i=1}^{7} w_i \, ln(x_i) \; + \lambda^T(c - Rx) \qquad - (3)$$

Forming the dual Problem

Maximize Lagrangian w.r.t. x

$$\frac{\partial \, \mathcal{L}(x,\lambda)}{\partial x_i} = \frac{w_i}{x_i} - (R^T.\lambda)_i = 0 \qquad - (4)$$

$$\Rightarrow \;\; x_i = \frac{w_i}{(R^T.\lambda)_i}$$

Therefore, the dual function is,

$$g(\lambda) = \sum_{i=1}^{7} w_i \, ln\left(\frac{w_i}{(R^T.\lambda)_i}\right) + \lambda^T c - \sum_{i=1}^{7} w_i \qquad -(5)$$

Dual Problem:

$$\min_{\lambda \geq 0} g(\lambda) \qquad -(6)$$

Dual Gradient Ascent Algorithm

1. Initialise $\lambda_0 \geq 0$
2. At iteration k :
   - Compute $x_i{}^k = \frac{w_i}{(R^T.\lambda^k)_i}$
   - Compute subgradient: $\nabla g(\lambda^k) = c - Rx^k$
   - Update:

     $$\lambda^{k+1} = [\lambda^k + \alpha_k(Rx^k - c)]_+$$

     where $[\cdot]_+$ denotes projection onto nonnegative orthant
   - Repeat until convergence

Problem 2 (b)

For 4000 iterations, and alpha = 0.01, Figure 23 represents the convergence of flow rates as a function of number of iterations.

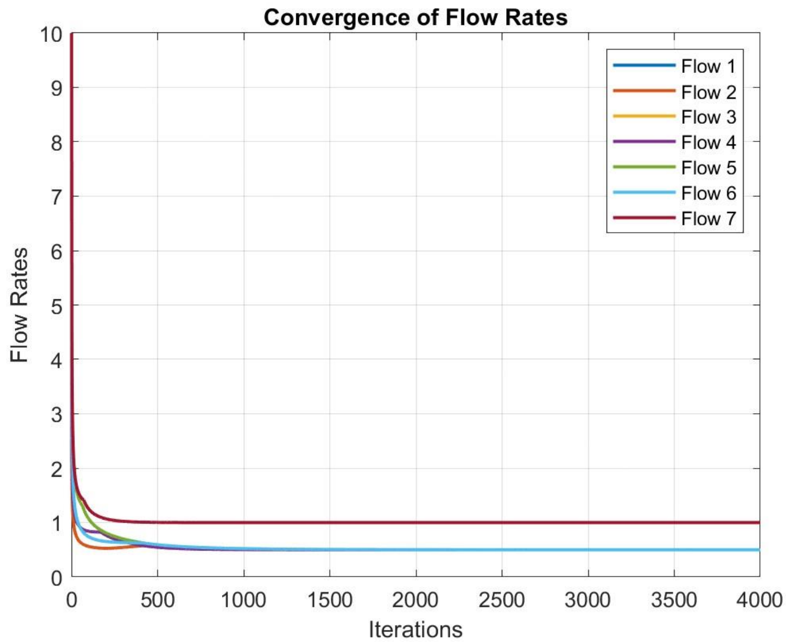

Figure 23

For 4000 iterations, and alpha = 0.01, Figure 24 represents the convergence of dual variables as a function of number of iterations.

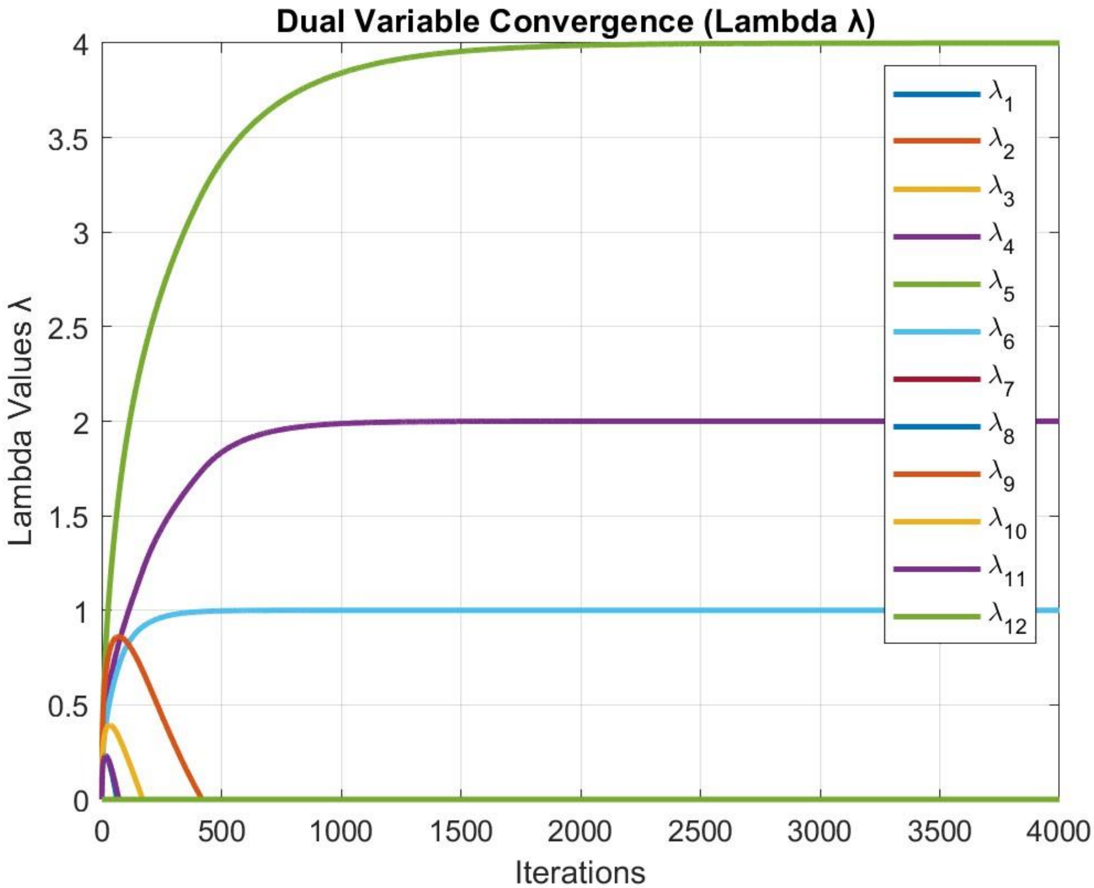

Figure 24

final_flows = { 0.5000, 0.5000, 0.5000, 0.5000, 0.5016, 0.5016, 1.0000}

Optimal Dual Variables = { 0.999999999174948, 0, 1.99991757386531,1.99991757386531, 3.98750261191152, 0.999999999174948, 0, 0, 0,0, 0, 0}

Optimal Dual Variables (approxed) = [1,0,2,2,4,1,0,0,0,0,0,0]

Problem 2 (c)

Current total rate at links: [1, 0, 1, 1, 1, 1, 0, 0.5, 1.5, 1, 1, 0]

Link capacity: [1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2]

Max constraint violation (Rx <= c): 2.09e-05

Min dual variable: 0.00e+00

Complementary slackness : 8.36e-05

Max stationarity error: 2.61e-08

Primal feasibility: true

Dual feasibility: true

Complementary slackness: true

Gradient of Lagrangian function: [-2.2204e-16, -2.2204e-16, -2.2204e-16, -2.2204e-16, 0, 0, 0]

Gradient of Lagrangian function is zero: true

KKT conditions satisfied - The given solution is optimal