UNIVERSITY OF TOULON

August 12, 2024

Optimisation Techniques

By:

Ishfaq Bhat

Under the supervision of: Prof.Francesca Chittaro

Consider the function

$$\begin{aligned}\phi(x) =& \frac{1}{2}\Big[4(x_3 - x_4)^2 + (2x_2 + x_3 - x_4)^2 + 8x_2 \\ &+ \frac{1}{4} - 80x_3 + 32x_4 + \frac{1}{7500}(x_1 + 2x_2 + x_3 + 4x_4)^2 \\ &+ \frac{1}{300} - 15(x_1 + 2x_2 + x_3 + 4x_4) + (x_1 + 2x_2 + x_3 + 4x_4)^2\Big].\end{aligned}$$

Using some version of the gradient descent algorithm, find the minimum of ϕ over the set $\Omega = \{x \in \mathbb{R}^4 : h_1(x) = h_2(x) = h_3(x) = 0\}$, where

$$\begin{aligned}h_1(x) &= x_1 - 2x_3 + x_4 - 1, \\ h_2(x) &= x_2 + x_3 - 2, \\ h_3(x) &= 2x_3 + 3x_4 - 1.\end{aligned}$$

Q1:

For each point, you must specify the algorithm you used (e.g. "gradient descent with optimal step", "projected gradient with fixed step", "Newton Raphson's"...), the stopping criterion you

Answer:

Optimization Results

In this problem, I employed two optimization algorithms: projected steepest descent (Gradient descent with optimal step) and projected gradient with fixed steps. The initial step size for both algorithms was set to $\alpha = 0.001$, $\delta = 0.005$ and $x_0 = (2, 1.5, 0.5, 0)$ (starting point given). Numerical differentiation was used to calculate the gradients. But i will only discuss the results Gradient descent with optimal step

Projected Steepest Descent Algorithm/Gradient descent with optimal step

For the projected steepest descent algorithm the convergence criterion used is $|\phi(x_{k+1}) - \phi(x_k)| < 10^{-10}$. The value of the conversion, set here is to satisfy the Lagrange's multipliers rule otherwise the function does not changes so much after 10^{-6} . After 324 iterations, the algorithm converged to the point where value of function is $\phi(x^*) = -14.0852$ at point $x^* = (-4.2417, 3.8406, -1.8406, 1.5604)$ (for calculation please see Matlab file Q_ONE.m)

Function($\phi(x^*)$ converges to -14.0852 and corresponding the values of the variables are [-4.2417, 3.8406, -1.8406, 1.5604] with 324 iterations.

Calculation

For calculations of gradients i used the centered differencing formula.

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Below is the formulas used in Matlab use for the numerical differentiation with respectx_1, x_2, x_3, x_4.

$$\frac{d\phi}{dx_1} = \frac{\phi(x_1+\delta, x_2, x_3, x_4) - \phi(x_1-\delta, x_2, x_3, x_4)}{2\delta}$$

$$\frac{d\phi}{dx_2} = \frac{\phi(x_1, x_2+\delta, x_3, x_4) - \phi(x_1, x_2-\delta, x_3, x_4)}{2\delta}$$

$$\frac{d\phi}{dx_3} = \frac{\phi(x_1, x_2, x_3+\delta, x_4) - \phi(x_1, x_2, x_3-\delta, x_4)}{2\delta}$$

$$\frac{d\phi}{dx_4} = \frac{\phi(x_1, x_2, x_3, x_4+\delta) - \phi(x_1, x_2, x_3, x_4-\delta)}{2\delta}$$

After the gradient calculation, we have to calculate the orthogonal projection matrix P which is calculate by:

$$P = I_n - A^\top \cdot (A \cdot A^\top)^{-1} \cdot A$$

where A is the matrix of the equation of the constraints given in problem. The matrix A is given by:

$$A = \begin{bmatrix} 1 & 0 & -2 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 3 \end{bmatrix}$$

In order to calculate values of X of the

$$x_{\text{new}} = x - \alpha P \nabla \phi$$

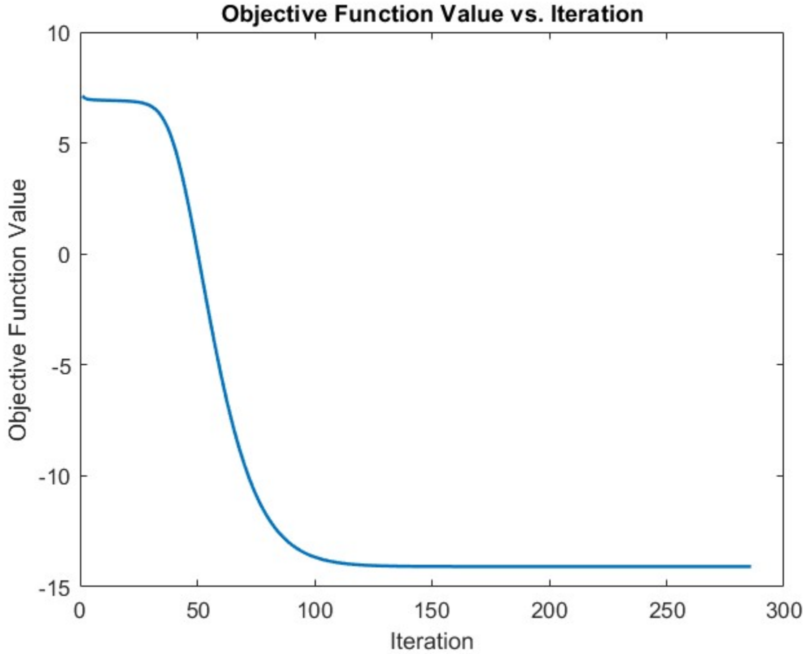
The optimal value of α can be calculated using the Nelder-Mead simplex method that for this we have in-built function in Matlab as **fminsearch**:

$$\alpha = \text{fminsearch}(\text{func}, 0)$$

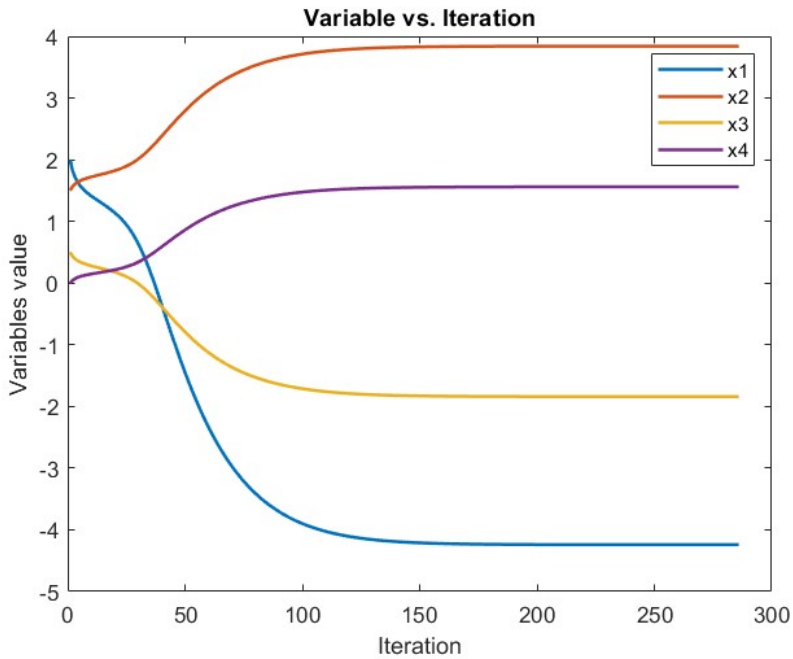
where func $= \phi(x - \alpha \cdot \text{grad\_phi})$, and α will be updated in the above equation

The stopping criteria used for this problem is $|\phi(x_{k+1}) - \phi(x_k)| < 10^{-10}$

Q1(b): **Draw a plot putting in abscissa the step of the algorithm and in ordinate the value of the function at the step of the algorithm.**



Below plot shows the values variables x_1, x_2, x_3, x_4 with respect iteration



Q1(c): **Prove (numerically or analytically) that the point founded at question 1. satisfies Lagrange's multipliers rule.**

Answer: In order to prove the above point in **Q1** our Lagrangian function.

$\mathcal{L}(x, \lambda) = \phi(x) + \lambda_1 \cdot h_1 + \lambda_2 \cdot h_2 + \lambda_3 \cdot h_3$.

and Lagrangian conditions are :

$$\nabla\phi + \lambda \cdot \nabla h(x) = 0$$

$$h(x) = 0$$

we calculated the gradient with respect to $x_1, x_2, x_3, x_4, \lambda_1, \lambda_2, \lambda_3$.

All these derivative's where calculated by as **symbolic derivatives** using the inbuilt function **diff**. The partial derivatives of L with respect to x_1, x_2, x_3, and x_4 λ_1, λ_2, and λ_3 are:

$$\begin{aligned}\frac{d\mathcal{L}}{dx_1} &= \lambda_1 + \frac{(x_1 + 2x_2 + x_3 + 4x_4)^2 \cdot (2x_1 + 4x_2 + 2x_3 + 8x_4 - 15)}{15000} \\ &- \frac{(2x_1 + 4x_2 + 2x_3 + 8x_4) \cdot (15x_1 + 30x_2 + 15x_3 + 60x_4 - (x_1 + 2x_2 + x_3 + 4x_4)^2 - 300)}{15000} = 0\end{aligned}$$

$$\begin{aligned}\frac{d\mathcal{L}}{dx_2} &= \lambda_2 + 4x_2 + 2x_3 - 2x_4 \\ &+ \frac{(x_1 + 2x_2 + x_3 + 4x_4)^2 \cdot (4x_1 + 8x_2 + 4x_3 + 16x_4 - 30)}{15000} \\ &- \frac{(4x_1 + 8x_2 + 4x_3 + 16x_4) \cdot (15x_1 + 30x_2 + 15x_3 + 60x_4 - (x_1 + 2x_2 + x_3 + 4x_4)^2 - 300)}{15000} = 0\end{aligned}$$

$$\begin{aligned}\frac{d\mathcal{L}}{dx_3} &= \lambda_2 - 2\lambda_1 + 2\lambda_3 + 2x_2 + 5x_3 - 5x_4 \\ &+ \frac{(x_1 + 2x_2 + x_3 + 4x_4)^2 \cdot (2x_1 + 4x_2 + 2x_3 + 8x_4 - 15)}{15000} \\ &- \frac{(2x_1 + 4x_2 + 2x_3 + 8x_4) \cdot (15x_1 + 30x_2 + 15x_3 + 60x_4 - (x_1 + 2x_2 + x_3 + 4x_4)^2 - 300)}{15000} = 0\end{aligned}$$

$$\begin{aligned}\frac{d\mathcal{L}}{dx_4} &= \lambda_1 + 3\lambda_3 - 2x_2 - 5x_3 + 13x_4 \\ &+ \frac{(x_1 + 2x_2 + x_3 + 4x_4)^2 \cdot (8x_1 + 16x_2 + 8x_3 + 32x_4 - 60)}{15000} \\ &- \frac{(8x_1 + 16x_2 + 8x_3 + 32x_4) \cdot (15x_1 + 30x_2 + 15x_3 + 60x_4 - (x_1 + 2x_2 + x_3 + 4x_4)^2 - 300)}{15000} \\ &- 120x_4^2 + 64x_4^3 = 0\end{aligned}$$

$$\frac{d\mathcal{L}}{d\lambda_1} = x_1 - 2x_3 + x_4 - 1 = 0$$

$$\frac{d\mathcal{L}}{d\lambda_2} = x_2 + x_3 - 2 = 0$$

$$\frac{d\mathcal{L}}{d\lambda_3} = 2x_3 + 3x_4 - 1 = 0$$

Now we have six unknown and six equations which can be solved with the help of Matlab using the in built function **solve**. After solving, our values of the $x_1, x_2, x_3, x_4, \lambda_1, \lambda_2, \lambda_3$ are **-4.2417 , 3.8407,-1.8407, 1.5604,-0.2577 ,-9.0759 , 8.8134** respectively (for calculation please see Matlab file with name Lagrange_1.m. If we compare values of x_1, x_2, x_3, x_4 from Lagrangian multiplier with what we calculated with projected gradient method in **Q1(a)** are same .

Q2:

Using some version of the gradient descent algorithm, find the minimum of ϕ over the set $\Upsilon = \{x \in \mathbb{R}^4 : h_4(x) = h_5(x) = 0\}$, where

$$h_4(x) = x_1 + x_2 + x_4 - 1,$$

$$h_5(x) = 5x_2 - 5x_4 - 1.$$

Use $x_0 = (1, 0.1, 1, -0.1)$ as the starting point. To which point does the algorithm converge? In how many steps? What is the value of the function at that point?

Answer:

In this problem, I used same algorithms as in above **Q1(a)**: Projected steepest descent algorithm. The initial step size for both algorithms was set to $\alpha = 0.001$, $\delta = 0.005$ and $x_0 = (1, 0.1, 1, -0.1)$ (starting point given) only constraints changes (h_4 and h_5). Numerical differentiation was used to calculate the gradients.

Projected Steepest Descent Algorithm

For the projected steepest descent algorithm the convergence criterion is $|\phi(x_{k+1}) - \phi(x_k)| < 10^{-10}$. After 39 iterations, the algorithm converged to the point where value of function is $\phi(x^*) = 0.0661$ at point x_1, x_2, x_3, x_4 **are 0.8936,0.1532,-0.1148,-0.0468)** (for calculation please see Matlab file Q_Second.m) For calculations of gradients I use the centered differencing formula.

Calculation

For calculations of gradients i used the centered differencing formula same as in $Q.1$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Below is the formulas used in Matlab use for the numerical differentiation with respect $[x_1, x_2, x_3, x_4]$.

$$\frac{d\phi}{dx_1} = \frac{\phi(x_1 + \delta, x_2, x_3, x_4) - \phi(x_1 - \delta, x_2, x_3, x_4)}{2\delta}$$

$$\frac{d\phi}{dx_2} = \frac{\phi(x_1, x_2 + \delta, x_3, x_4) - \phi(x_1, x_2 - \delta, x_3, x_4)}{2\delta}$$

$$\frac{d\phi}{dx_3} = \frac{\phi(x_1, x_2, x_3 + \delta, x_4) - \phi(x_1, x_2, x_3 - \delta, x_4)}{2\delta}$$

$$\frac{d\phi}{dx_4} = \frac{\phi(x_1, x_2, x_3, x_4 + \delta) - \phi(x_1, x_2, x_3, x_4 - \delta)}{2\delta}$$

After the gradient calculation, we have to calculate the orthogonal projection matrix P which is calculate by:

$$P = I_n - A^\top \cdot (A \cdot A^\top)^{-1} \cdot A$$

where A is the matrix of the equation of the constraints given in problem(h_4 and h_5). The matrix A is given by:

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 5 & 0 & -5 \end{bmatrix}$$

In order to calculate values of X of the

$$x_{\text{new}} = x - \alpha P \nabla \phi \tag{1}$$

The equation to find the optimal α using the Nelder-Mead simplex method that is in-built function in Matlab as **fminsearch**:

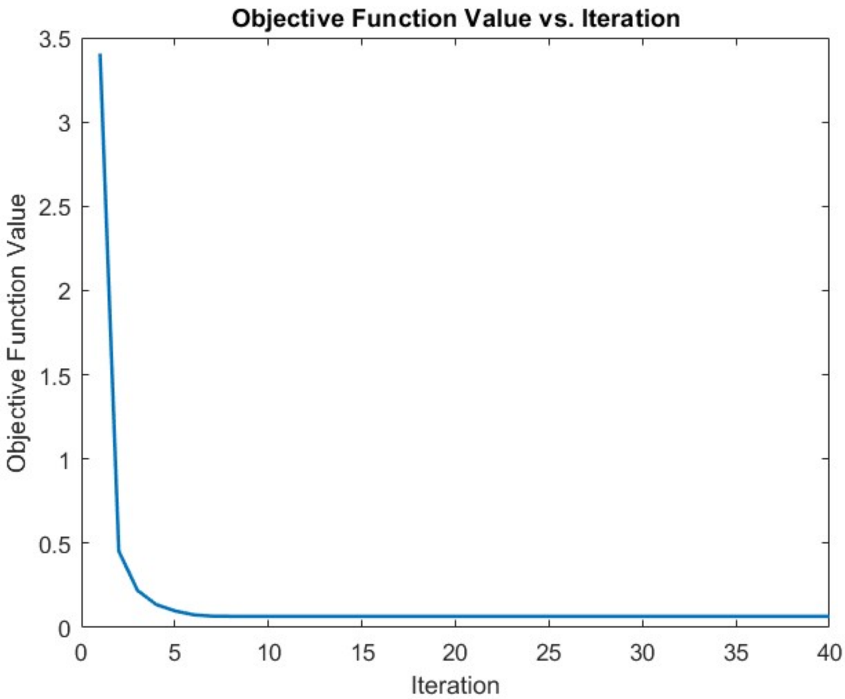
$$\alpha = \text{fminsearch}(\text{func}, 0) \tag{2}$$

where func $= \phi(x - \alpha \cdot \text{grad\_phi})$.

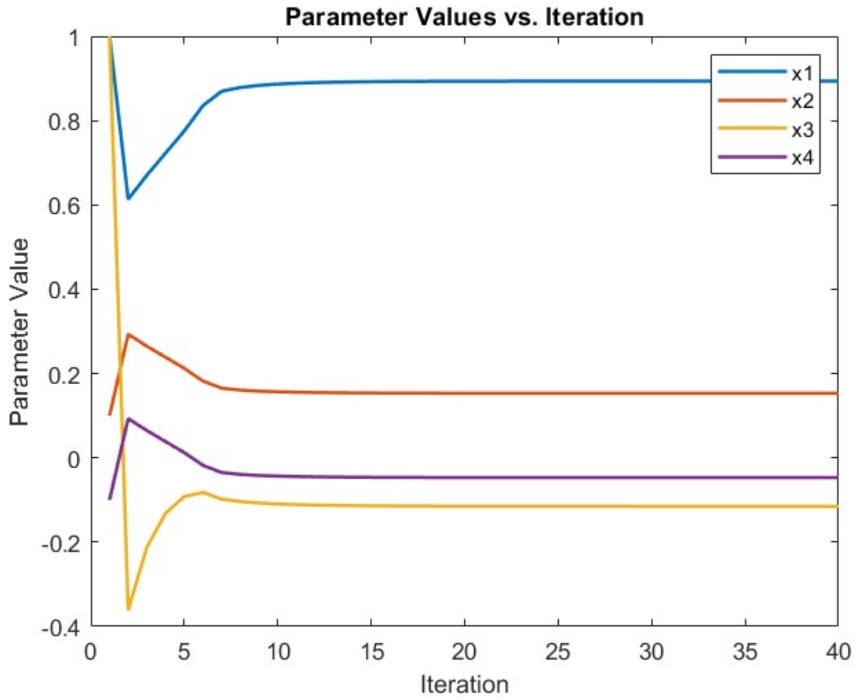
The stopping criteria used for this problem is $|\phi(x_{k+1}) - \phi(x_k)| < 10^{-10}$.

Q.2(b)

Draw a plot putting in abscissa the step of the algorithm and in ordinate the value of the function at the step of the algorithm



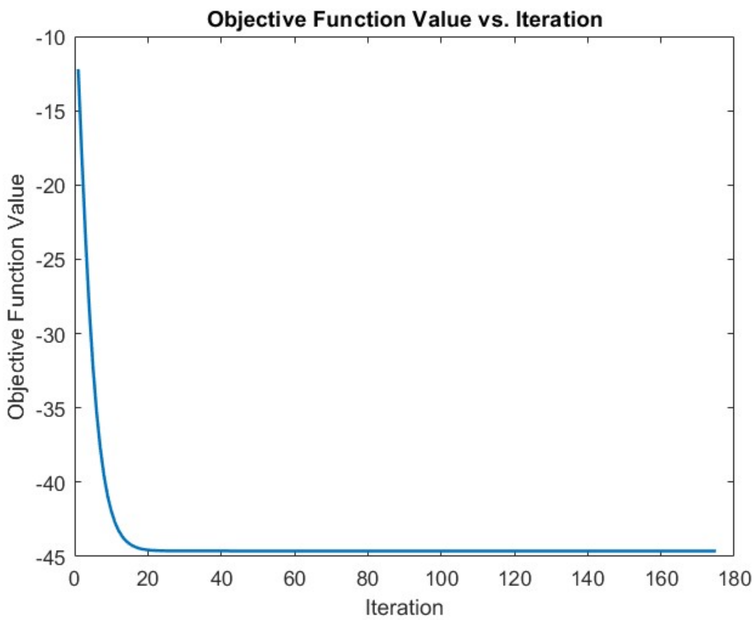
Below is the plot between iteration and variables x_1, x_2, x_3, x_4

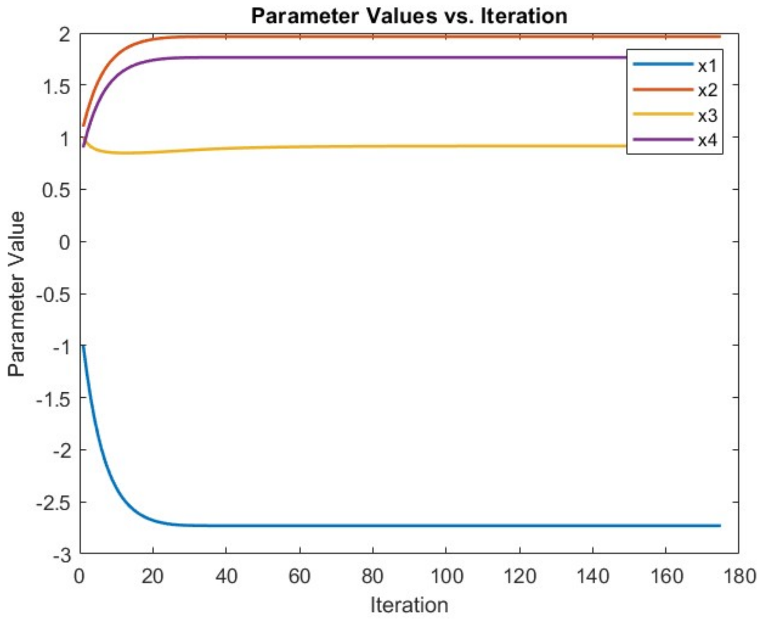


Q.3 Now repeat points 2.a)-b) with starting point $x_0 = (-1, 1.1, 1, 0.9)$. How do you explain your results

Answer:a

Below are the plots of objective function vs iteration and variables vs iteration with starting point x_0 given in question. I used the same code and method only starting point changes.





Answer:b

From the above graph it is obvious that function has the local minimum, if we set initial point as given in **Q.2** we converge at $\phi(x^*) = 0.0661$, while in **Q.3** our objective function starts below this point and converges to **-44.6281** after **174** iteration with x_1, x_2, x_3, x_4 equal to **-2.7305, 1.9653, 0.9150,1.7653** with same $\alpha = 0.001$, $\delta = 0.005$ as used in **Q.1** and **Q.2**. Form this it is confirmed that function have multiple local minimum. In order to find the global minimum we have to evaluate this function on multiple starting point to see where our function converges.

Additional

For this calculation I deduct that Lagrange multipliers method is not the optimal way to find the minimum of the function. Lagrange multiplier method finds critical points where the gradient of the objective function is orthogonal to the constraints but not necessary the global one.

Prove (numerically or analytically) that the point founded at question 2 satisfies Lagrange's multipliers rule for point Using the Lagrange multiplier rule we have we have Lagrangian function.

Calculation

$\mathcal{L}(x,\lambda) = \phi(x) + \lambda_1 \cdot h_4 + \lambda_2 \cdot h_5$

$$\nabla\phi + \lambda \cdot \nabla h(x) = 0$$

$$h(x) = 0$$

we will calculate the gradient with respect to $x_1, x_2, x_3, x_4, \lambda_1, \lambda_2$

$$\begin{aligned}\frac{d\mathcal{L}}{dx_1} = \lambda_1 + &\frac{(x_1+2x_2+x_3+4x_4)^2(2x_1+4x_2+2x_3+8x_4-15)}{15000} \\ &- \frac{(2x_1+4x_2+2x_3+8x_4)(15x_1+30x_2+15x_3+60x_4-(x_1+2x_2+x_3+4x_4)^2-300)}{15000} = 0\end{aligned}$$

$$\begin{aligned}\frac{d\mathcal{L}}{dx_2} = \lambda_1 &+ 5\lambda_2 + 4x_2 + 2x_3 - 2x_4 \\ &+ \frac{(x_1+2x_2+x_3+4x_4)^2(4x_1+8x_2+4x_3+16x_4-30)}{15000} \\ &- \frac{(4x_1+8x_2+4x_3+16x_4)(15x_1+30x_2+15x_3+60x_4-(x_1+2x_2+x_3+4x_4)^2-300)}{15000} = 0\end{aligned}$$

$$\begin{aligned}\frac{d\mathcal{L}}{dx_3} = 2x_2 + 5x_3 - 5x_4 + &\frac{(x_1+2x_2+x_3+4x_4)^2(2x_1+4x_2+2x_3+8x_4-15)}{15000} \\ &- \frac{(2x_1+4x_2+2x_3+8x_4)(15x_1+30x_2+15x_3+60x_4-(x_1+2x_2+x_3+4x_4)^2-300)}{15000} = 0\end{aligned}$$

$$\begin{aligned}\frac{d\mathcal{L}}{dx_4} = \lambda_1 &- 5\lambda_2 - 2x_2 - 5x_3 + 13x_4 \\ &+ \frac{(x_1+2x_2+x_3+4x_4)^2(8x_1+16x_2+8x_3+32x_4-60)}{15000} \\ &- \frac{(8x_1+16x_2+8x_3+32x_4)(15x_1+30x_2+15x_3+60x_4-(x_1+2x_2+x_3+4x_4)^2-300)}{15000} \\ &- 120x_4^2 + 64x_4^3 = 0\end{aligned}$$

$$\frac{d\mathcal{L}}{d\lambda_1} = x_1 + x_2 + x_4 - 1 = 0$$

$$\frac{d\mathcal{L}}{d\lambda_2} = 5x_2 - 5x_4 - 1 = 0$$

By solving these equation with help of Matlab code (for code please see Lagrange_2.m file in attachment) we get the values of $x_1, x_2, x_3, x_4, \lambda_1, \lambda_2$ as 0.8937 **0.1531 -0.1149 -0.0469 -0.0337 -0.1021** are same as what we got in **Q.2** when we use $x_0 = (1, 0.1, 1, -0.1)$ as the starting point