# ChE 597 Computational Optimization

## Homework 5 Solutions

1. Consider the following linear program where the linear equalities and inequalities are separated.

$$\begin{aligned} \min_x \; & c^T x \\ \text{s.t. } & Ax = b \\ & Gx \leq h \end{aligned}$$

   (a) Write down the KKT conditions for this problem.

   (b) Compare the KKT conditions with the results we got from linear programming optimality conditions in the LP duality lecture (Lecture 7). What can you observe? Are they the same?

**Solution: PART A)**

For the given system, let $x \in \mathbb{R}^n, A \in \mathbb{R}^{p\times n}$ and $G \in \mathbb{R}^{m\times n}$, with $b$ and $h$ of suitable conformity.

Let objective $c^T x$ be represented as $f_0(x)$, with $Gx \leq h$ and $Ax = b$ be represented by following respective (in-) equalities $f_i(x) \leq 0 \;\; \forall i \in \{1,2,..,m\}$ and $h_i(x) = 0 \;\; \forall i \in \{1,2,..,p\}$, with domain of the system being $x \in D$.

The lagrangian then would be given as:
$L(x,\lambda,\nu) = \nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + \sum_{i=1}^p \nu_i \nabla h_i(x)$

Hence, the dual function:
$g(\lambda,\nu) = min_{x\in D}\, L(x,\lambda,\nu)$

For some feasible points, $\hat{x}, \hat{\lambda}$ and $\hat{\nu}$, KKT conditions would be given as:

$$\begin{aligned} &\text{Primal Feasibility} \\ f_i(\hat{x}) &\leq 0, && i = 1,\ldots,m \\ h_i(\hat{x}) &= 0, && i = 1,\ldots,p \\ &\text{Dual Feasibility} \\ \hat{\lambda}_i &\geq 0, && i = 1,\ldots,m \\ &\text{Complementary Slackness} \\ \hat{\lambda}_i f_i(\hat{x}) &= 0, && i = 1,\ldots,m \\ &\text{Stationarity} \\ \nabla f_0(\hat{x}) + \sum_{i=1}^m \hat{\lambda}_i \nabla f_i(\hat{x}) &+ \sum_{i=1}^p \hat{\nu}_i \nabla h_i(\hat{x}) = 0, \end{aligned}$$

Now, since the primal problem along with the constraints are convex, any pair of primal and dual feasible points $[\hat{x}, (\hat{\lambda}, \hat{\nu})]$ that satisfy KKT conditions would also be primal and

dual optimal with zero duality gap. Therefore, the conditions serve to be sufficient for optimality.

For problem system with given notations, the equations translate to:

$$\begin{aligned}
&\text{Primal Feasibility} \\
&G\hat{x} \leq h \\
&A\hat{x} = b \\
&\text{Dual Feasibility} \\
&\hat{\lambda} \geq 0 \\
&\text{Complementary Slackness} \\
&\hat{\lambda}_i(G_i\hat{x} - h_i) = 0, \qquad i = 1,\ldots,m \\
&\text{Stationarity} \\
&c + G^T\hat{\lambda} + A^T\hat{\nu} = 0,
\end{aligned}$$

Here, $G_i$ represents $i^{th}$ row vector.

**PART B)**

We first independently observe the optimality conditions using LP duality. The dualised function can be written as:

$min_x\ g(\lambda,\nu) = min_x\ c^Tx + \lambda^T(Gx-h) + \nu^T(Ax-b)$

Where, $\lambda \geq 0$

$\implies\ -\lambda^T h - \nu^T b + min_x\ (c + G^T\lambda + A^T\nu)^T x$

Now, for $\lambda$ and $\nu$ to be dual feasible, we must have $(c + G^T\lambda + A^T\nu)^T = 0$

Therefore, combining primal and dual feasibility gives, for some $[\hat{x},(\hat{\lambda},\hat{\nu})]$ :

$G\hat{x} \leq h$

$A\hat{x} = b$

$\hat{\lambda} \geq 0$

$c + G^T\hat{\lambda} + A^T\hat{\nu} = 0$

Finally, the requirement of 0 duality gap gives us the complementary slackness condition:

$\hat{\lambda}_i(G_i\hat{x} - h_i) = 0,\ \ i = 1,\ldots,m$

Hence, we conclude to get the same set of conditions as of using KKT conditions.

2. Consider the following problem with a quadratic objective (without linear and constant term) subject to one quadratic and $m$ linear inequalities constraints.

$$\begin{aligned} &\min_x x^T Q^0 x \\ \text{s.t. } &x^T Q^1 x + q^T x + r \leq 0 \\ &Ax \leq b \end{aligned}$$

where $Q^0$ and $Q^1$ can be any symmetric matrices (they are not necessarily PSD).

(a) Derive the Lagrangian dual function.

(b) Derive the dual maximization problem as SDP using Schur's lemma.

**Solution: PART A)**
The dualised problem can be written as:
$L(x,\lambda,\nu) = x^T Q^0 x + \lambda(x^T Q^1 x + q^T x + r) + \nu^T(Ax-b)$ Where, $\lambda, \nu \geq 0$ of conformable dimensions.
Dual function is given as:
$g(\lambda,\nu) = min_x \; x^T Q^0 x + \lambda(x^T Q^1 x + q^T x + r) + \nu^T(Ax-b)$
Dual Problem:
$max_{\lambda,\nu \geq 0} \; g(\lambda,\nu)$

**PART B)**
For a feasible dual problem, we must have the QP: $min_x \; x^T Q^0 x + \lambda(x^T Q^1 x + q^T x + r) + \nu(Ax-b)$ bounded. By re-arranging we can write:

$g(\lambda,\nu) = min_x \; x^T(Q^0 + \lambda Q^1)x + (\lambda q^T + \nu^T A)x + \lambda r - \nu^T b$
Let's define:
$Q'(\lambda,\nu) = Q^0 + \lambda Q^1$
$q'(\lambda,\nu) = q + A^T\nu$
$r'(\lambda,\nu) = \lambda r - \nu^T b$

Following which, we can compactly write the conditions required for boundedness of the quadratic problem, i.e. 1. $Q'(\lambda,\nu) \succeq 0$
2. $q'(\lambda,\nu) = \mathcal{R}(Q'(\lambda,\nu))$

More concisely, we have (as discussed in Lecture 9):

$$g(\lambda,\nu) = \begin{cases} -\frac{1}{4} q'(\lambda,\nu)^T Q'^{\dagger}(\lambda,\nu) q(\lambda,\nu) + r'(\lambda,\nu), & \text{if } q'(\lambda,\nu) \in \mathcal{R}\big(Q'(\lambda,\nu)\big) \\ & \text{and } Q'(\lambda,\nu) \succeq 0 \\ -\infty, & \text{otherwise} \end{cases}$$

Now using Schur's lemma, the dual maximisation problem, $max_{\lambda,\nu \geq 0} \; g(\lambda,\nu)$ can be written as: $max_{\lambda,\nu \geq 0, d} \; d$

$$\begin{bmatrix} r'(\lambda,\nu) - d & \frac{1}{2} q'(\lambda,\nu)^\top \\ \frac{1}{2} q'(\lambda,\nu) & Q'(\lambda,\nu) \end{bmatrix} \succeq 0.$$

3. Consider the nonlinear programming problem

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } g_j(x) \leq 0 \quad j = 1\ldots,r \\ &x \in R^n \end{aligned}$$

where the functions $f$ and $g$ are monotone in each variable $x_i$ (i.e. $\partial \mathrm{f}/\partial \mathrm{x_i}$ and $\partial \mathrm{g_j}/\partial \mathrm{x_i}$ are one-signed if the derivatives are non-zero).

Show that the following holds true for the optimal solutions of this problem if the problem satisfies some constraint qualification (these are the so called principles of monotonicity analysis):

(a) If a variable $x_i$ is present in the objective and in some of the constraints, there is at least one active constraint involving $x_i$ and whose derivative has opposite sign from the objective.

(b) If a variable $x_i$ is not present in the objective, it is either involved in at least two active constraints with opposite sign in the derivatives, or else it is not involved in any active constraints.

**Solution:** For the given system, the dualized function can be written as:

$L(x,\lambda) = f(x) + \sum_j \lambda g_j \; ; \; \lambda \geq 0$

The dual problem for the dual function $g(\lambda)$, is hence given by:
$max_\lambda g(\lambda) = max_\lambda \; min_x \; f(x) + \lambda g_j$

Now, for the case of optimal solution i.e. having strong duality under some constraint qualification. We must have KKT conditions satisfied as learnt in Lecture 9. Hence, we go on to write:

- $g_j(\hat{x}) \leq 0$
- $\hat{\lambda}_j \geq 0$
- $\hat{\lambda}_j g_j(\hat{x}) = 0$
- $\nabla f + \sum_j \hat{\lambda}_j \nabla g_j = 0$

The last condition can be written as:
$\frac{\partial f}{\partial x_i} + \sum_j \hat{\lambda}_j \frac{\partial g_j}{\partial x_i} = 0 \quad \in i = 1,2,...,r$

**Part a)**

Now, if variable $x_i$ is present in objective, we must have: $\frac{\partial f}{\partial x_i} \neq 0$

$\implies \sum_j \hat{\lambda}_j \frac{\partial g_j}{\partial x_i} = -\frac{\partial f}{\partial x_i}$

Now, let set of active constraints be represented by $j \in J$, then we must have:
$g_j(x) = 0$
implying
$\lambda_j \geq 0 \;\; \forall j \in J$, since we must have $\hat{\lambda}_j g_j(\hat{x}) = 0$. Note that for all the inactive constraints

we must have $\lambda_j = 0$.

$\implies \sum_{j \in J} \hat{\lambda}_j \frac{\partial g_j}{\partial x_i} = -\frac{\partial f}{\partial x_i}$
Now, if the set is empty, that would yield, $0 = -\frac{\partial f}{\partial x_i}$, which would be a contradiction to the condition with which we started with, hence the set should be non-empty i.e. at least for one of the $j's$ we would have $\hat{\lambda}_j \frac{\partial g_j}{\partial x_i} \neq 0$

Now, let all the active derivatives have the same sign as objective, then since $\hat{\lambda}_j \geq 0$, the combination $\sum_j \hat{\lambda}_j \frac{\partial g_j}{\partial x_i}$ would also have same sign as of the objective's derivative. However, this would imply the combination's sum to be equal to a value of opposite sign, yielding a contradiction. Therefore, we must have at-least one active constraint involving $x_i$ and whose derivative has opposite sign from the active $x_i$ in the objective.

**Part b)**

If the variable $x_i$ is not present in the objective, then we would be having $\frac{\partial f}{\partial x_i} = 0$, yielding $\sum_j \hat{\lambda}_j \frac{\partial g_j}{\partial x_i} = 0$

For the involved variables $x_i$, let again the set of active constraints be represented by $j \in J$, then we must have: $g_j(x) = 0$ implying
$\lambda_j \geq 0$, since we must have $\hat{\lambda}_j g_j(\hat{x}) = 0$.

Now, if the variable $x_i$ is not involved in any active constraint, then corresponding multiplier $\hat{\lambda}_j = 0$ and we would have $\sum_j \hat{\lambda}_j \frac{\partial g_j}{\partial x_i} = 0$ hold trivially.

On the other hand, if it is involved and for at least one $j$ we have, $\lambda_j \frac{\partial g_j}{\partial x_i} \neq 0$. However, if the active set has only one element, then the expression $\sum_{j \in J} \hat{\lambda}_j \frac{\partial g_j}{\partial x_i} = 0$ cannot hold true, therefore set cannot be singleton. Thus, we have established that if the variable is involved with one active constraint, then it has to be involved in at least two constraints.

Now, let all the active derivatives have the same sign, then since $\hat{\lambda}_j \geq 0$, the combination sum $\sum_j \hat{\lambda}_j \frac{\partial g_j}{\partial x_i}$ would also have the same sign combined and cannot be equal to zero. Thus, by contradiction we can say that $x_i$ has to be involved in at least two active constraints with opposite sign in the derivatives.

4. Consider the design of a storage vessel that has the form of a cylinder. The required volume is 25 $m^3$. The cost of the side of the cylinder is $\$150/m^2$, while the top and bottom cost $\$190/m^2$ and $260/m^2$ respectively. Formulate an NLP problem to determine the optimal dimensions of this vessel, and solve with Pyomo using the interior point solver IPOPT (check our pyomo installation guide if you have not installed IPOPT).

**Solution:** Let a cylinder be constructed of height h and radius r. Then, the required volume constraint of 25 $m^3$ mathematically translates to: $\pi r^2 h = 25$

The cost, C incurred to make such a cylinder would be:
$C = 150(2\pi rh) + 190\pi r^2 + 260\pi r^2$
$\implies C = 300\pi rh + 450\pi r^2$

Thus, we get the following Non linear optimisation problem:

$$min_{r,h} \; 300\pi rh + 450\pi r^2$$

$$\text{s.t.} \quad \pi r^2 h = 25$$

Solving with Pyomo (solver IPOPT) leads to the following optimal dimensions:
r = $1.384m$
h = $4.153m$
Objective (Minimum Cost) = 8126.987

The link to the Jupyter notebook is:
http://tinyurl.com/597HW5Q4

5. Consider the following convex optimization problem

$$\begin{aligned} \min_{x_1,x_2} \quad & \frac{3}{x_1+x_2} + e^{x_1} + (x_1-x_2)^2 \\ \text{s.t.} \quad & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_1^2 + x_2^2 \leq 2 \\ & x_1 - x_2 \leq 1 \end{aligned}$$

You will implement different versions of the interior point method to solve the problem. Use (0.5, 0.5) as the starting point for all the algorithms.

(a) Plot the feasible region using python.

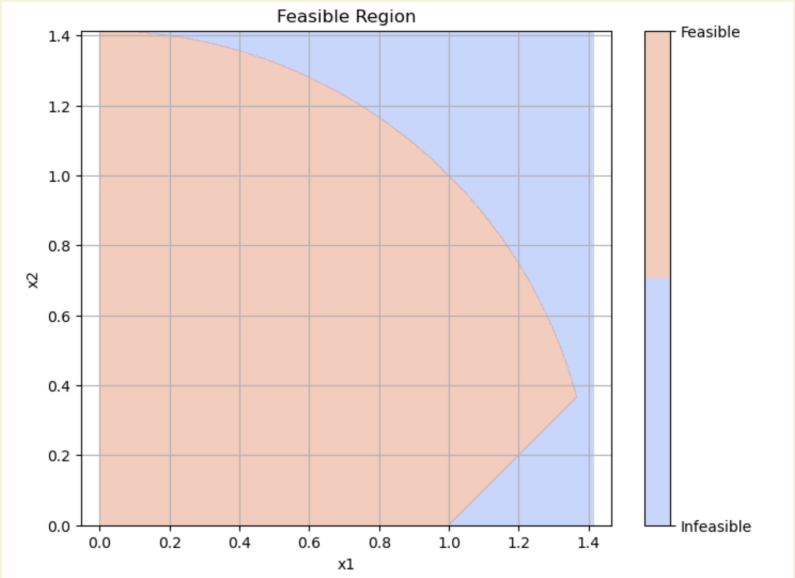
(b) Implement the barrier algorithm to solve the problem. Use $t^0 = 1$, $\mu = 4$, $\varepsilon = 10^{-4}$. For the Newton's method- use $\alpha = 0.2$, $\beta = 0.5$ and for convergence $\varepsilon_{newton} = 10^{-5}$. Also, plot path of the variables on the plot conceived in (a).

(c) Implement v1 of the primal dual interior point method. Use $t^0 = 1$, $\mu = 1.2$, $\varepsilon = 10^{-5}$. For backtracking, use $\alpha = 0.5$ and $\beta = 0.3$.

(d) Implement v2 of the primal dual interior point method. Use $u^0 = 1^T$, $\mu = 1.2$, $\varepsilon = 10^{-5}$. For backtracking, use $\alpha = 0.5$ and $\beta = 0.3$. Note: here $1^T = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T$

**Solution: PART A)**

The feasible region given by the set of inequalities is




Note that it is a convex region.

The link to the Jupyter notebook is:
http://tinyurl.com/HW5Q5a

**PART B)**

For the given function $f_o(x)$, $x \in \mathbb{R}^2$ having only constraining inequalities the Barrier function can be written as:

$min_x\, t f_o(x) + \phi(x)$;
having no constraint of any sort. Where, $\phi(x) = \sum_{i \in I} log(-f_i(x))$, set I representing

the inequalities defined by $f_i(x) \leq 0 \;\; i = 1,2,3,4$.

Now, since we don't have any constraint to our optimisation problem, we can use Newton's method with backtracking line search as done previously in Homework 2, Problem 1.

By defining our barrier function as:

$$B(x) = t f_o(x) + \phi(x)$$

**Newton's method** (damped/guarded, $t \neq 1$)

**Algorithm:**

**Input:** Starting point $x \in \mathrm{dom}\, B$ or some known initialisation, tolerance $\varepsilon_{newton} = 10^{-5}$

**Repeat:**

i. Compute the Newton step and decrement.
$\Delta x_{nt} := -\nabla^2 B(x)^{-1} \nabla B(x)$; $\lambda^2 := \nabla B(x)^T \nabla^2 B(x)^{-1} \nabla B(x)$.

ii. Stopping criterion. **quit** if $\lambda^2/2 \leq \varepsilon_{newton}$.

iii. Line search. Choose step size $t$ by backtracking line search

iv. Update. $x := x + t\Delta x_{nt}$.

The barrier algorithm is implemented as:

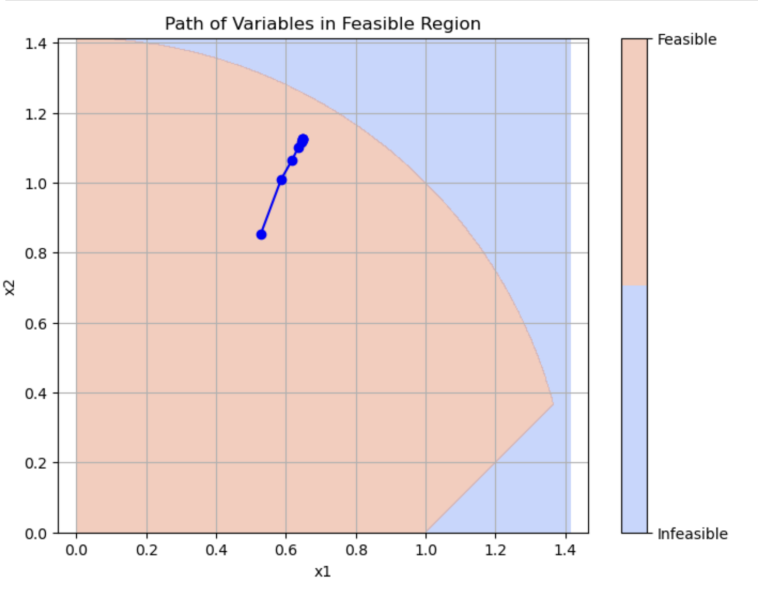
i. We fix $t^{(0)} = 1, \mu = 1.2$. We use Newton to compute $x^{(0)} = x^\star(t)$, solution to barrier problem at $t = t^{(0)}$.

ii. For $k = 1,2,3,\ldots$, solve the barrier problem at $t = t^{(k)}$, using Newton initialized at $x^{(k-1)}$, to yield $x^{(k)} = x^\star(t)$

iii. Stop if $m/t \leq \varepsilon = 10^{-4}$, else update $t^{(k+1)} = \mu t$

We get the below solution:
Optimal solution: [0.64728388 1.12486366]
Optimal function value: 3.831
Number of Newton iterations taken to converge: 8

The central path taken during the course of optmisation by x is given as:




The link to the Jupyter notebook is:
http://tinyurl.com/HW5Q5b

**PART C)**

Implementing the version 1 of Primal-Dual interior point method without lagrangian multipliers in the system.

Since, we don't have any equality constraint, function $r(x)$ or alternately $r_{dual}(x)$ is given as-

$$r_{dual}(x,t) \text{ or } r(x,t) = \left( \nabla f_o(x) + \sum_{i=1}^{m} \left( -\frac{1}{t f_i(x)} \right) \nabla f_i(x) \right) = 0$$

which is a pure function of x for any given t.

Hence, the newton direction can be found as:

$$r'(x)\Delta x = r(x)$$

where. where $r'(x) =$

$$\nabla^2 f_o(x) + \sum_{i=1}^{m} \frac{1}{t f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T + \sum_{i=1}^{m} \left( -\frac{1}{t f_i(x)} \right) \nabla^2 f_i(x)$$

**v1 Algorithm (without Equality Constraints):**

**Input:** start with $x^{(0)}$ such that $f_i(x^{(0)}) < 0$, $i = 1,\ldots,4,$. Define $t^{(0)} = 1$, we fix $\mu = 1.2$, repeat for $k = 1,2,3,\ldots$

**Repeat:**

i. Compute the direction $\Delta x$.

ii. Use backtracking to determine step size $s$

iii. Update $x^{(k)} = x^{(k-1)} + s \cdot \Delta x$

iv. Stop if $(\|r_{\text{dual}}\|^2)^{1/2} \leq \varepsilon$ else update $t^{(k+1)} = \mu t^{(k)}$

Next, we state the algorithm used for backtracking:

**Backtracking Function:**

**Input:** Define parameters $\alpha, \beta \in (0,1)$ as given, we set $s = 0.999$

**Function:**

i. Start with s = 0.999

ii. Perform update as: $x^+ = x + s\Delta x$

iii. Let $s = \beta s$, until $f_i(x^+) < 0, i = 1, \ldots, m$

iv. Let $s = \beta s$, until $\|r(x^+)\|_2 \leq (1 - \alpha s)\|r(x)\|_2$

Finally, we use a trick to ease up some calculations specific to this problem. Define:
$g(x,y) = f_0(x) - \sum_{i=1}^{4} \frac{f_i(x)}{t f_i(y)}$
with this, we can write:
$\nabla g_x(x,x) = r(x)$
and get couple of terms of $r'(x)$ because:
$\nabla^2 g_{xx}(x,x) = r'(x) - \sum_{i=1}^{m} \frac{1}{t f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T$
Thus, we can define $r'(x)$ as:
$r'(x) = \nabla^2 g_{xx}(x,x) + \sum_{i=1}^{m} \frac{1}{t f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T$

This helps us cut some steps while writing the code. The solution upon using this implementation is:
Optimal solution: [0.647 1.124]
Optimal function value: 3.831
Number of iterations taken to converge: 41

The link to the Jupyter notebook is:
http://tinyurl.com/HW5Q5c

**PART D)**

Implementing the version 2 of Primal-Dual interior point method with lagrangian multipliers in the system.

Since, again we don't have any equality constraint, we define $r_{dual}(x)$ and $r_{cent}(x)$ as:

$$r_{\text{dual}} = \nabla f_0(x) + Df(x)^T u \in \mathbb{R}^{2\times 1}$$

$$r_{\text{cent}} = -\text{diag}(u) f(x) - (1^T) t \in \mathbb{R}^{4\times 1}$$

$$r_{\text{prim}} = N.A.$$

where,

$$f(x) = \begin{bmatrix} f_1(x) \\ \ldots \\ f_4(x) \end{bmatrix} \in \mathbb{R}^{4\times 1}$$

and

$$Df(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \ldots \\ \nabla f_4(x)^T \end{bmatrix} \in \mathbb{R}^{4\times 2}$$

Now, root-finding update $\Delta y = (\Delta x; \Delta u) \in \mathbb{R}^{6\times 1}$ is given by

$$\begin{bmatrix} H_{\text{pd}}(x) & Df(x)^T \\ -\text{diag}(u)Df(x) & -\text{diag}(f(x)) \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta u \end{pmatrix} = - \begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \end{pmatrix}$$

where $H_{\text{pd}}(x) = \nabla^2 f_0(x) + \sum_{i=1}^{m} u_i \nabla^2 f_i(x)$

$r$ can be defined as:

$$r = \begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \end{pmatrix} \in \mathbb{R}^{6\times 1}$$

and $r'$ as:

$$r' = \begin{bmatrix} H_{\text{pd}}(x) & Df(x)^T \\ -\text{diag}(u)Df(x) & -\text{diag}(f(x)) \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

With the dimensions noted, one can check for the matrix conformity as well at each point.

**v2 Algorithm (without Equality Constraints):**

**Input:** Start with given $x^{(0)}$ (such that $f_i(x^{(0)}) < 0$, $i = 1,\ldots,4$), and given $u^{(0)}$. Define $\eta^{(0)} = -f(x^{(0)})^T u^{(0)}$ and $\varepsilon = 10^{-5}$ . We fix $\mu = 1.2$, repeat for $k = 1,2,3,\ldots$

**Repeat:**

i. Define $t = \mu m/\eta^{(k-1)}$ (Note, m = No. of inequalities = 4)

ii. Compute primal-dual update direction $\Delta y$

iii. Use backtracking to determine step size $s$

iv. Update $y^{(k)} = y^{(k-1)} + s \cdot \Delta y$

v. Compute $\eta^{(k)} = -f(x^{(k)})^T u^{(k)}$

vi. Stop if $\eta^{(k)} \leq \varepsilon$ and $(\|r_{\text{dual}}\|^2)^{1/2} \leq \varepsilon$

Next, the backtracking algorithm is given as:

**Backtracking Function:**

**Input:** Define parameters $\alpha, \beta \in (0,1)$ as given, we set $s_{max} = 1$. Further, take $y, \Delta y$ and $t$ as parameters in the function.

**Function:**

i. Having $s_{\max} = 1$, update it as to make $u + s\Delta u \geq 0$ :
$s_{\max} = \min\{1, \min\{-u_i/\Delta u_i : \Delta u_i < 0\}\}$

ii. Perform update as: $y^+ = y + s\Delta y$ and extract $x^+$ and $u^+$

iii. We set $s = 0.999 s_{\max}$

iv. Do $s = \beta s$, until $f_i(x^+) < 0, i = 1,\ldots,4$

v. Do $s = \beta s$, until $\|r(x^+, u^+)\|_2 \leq (1 - \alpha s)\|r(x,u)\|_2$

After, performing the optimisation, we get:
Optimal solution: [[0.647] [1.125]]
Optimal function value: [3.831]

Number of iterations taken to converge: 241

The link to the Jupyter notebook is:
http://tinyurl.com/HW5Q5d