098311 Optimization 1 Spring 2018
HW 10

Chen Tessler 305052680
Orr Krupnik 302629027

June 24, 2018

Solution 1.

$$\begin{aligned} &\min -2x_1^2 + 2x_2^2 + 4x_1 \\ &\text{s.t. } x_1^2 + x_2^2 - 4 \leq 0 \\ &\quad\ (x_1-2)^2 + x_2^2 - 1 \leq 0 \end{aligned}$$

1. The first constraint defines a compact set, and specifically the closed ball $B[(0,0),2]$ (i.e. a ball centered around 0 with radius 2). The second constraint defines the set $B[(2,0),1]$ (which is also a compact set). The intersection of these two circles is a non-empty compact set.

   Since the objective function is continuous over a compact set, we know (Weirstrass) that there exists an optimal solution to the problem.

2. The Lagrangian is:

   $$L(\lambda, x) = -2x_1^2 + 2x_2^2 + 4x_1 + \lambda_1(x_1^2 + x_2^2 - 4) + \lambda_2(x_1^2 + x_2^2 - 4x_1 + 3)$$

   comparing the gradient to zero results in:

   $$\nabla_x L = \begin{pmatrix} -4x_1 + 4 + 2\lambda_1 x_1 + 2\lambda_2 x_1 - 4\lambda_2 \\ 4x_2 + 2\lambda_1 x_2 + 2\lambda_2 x_2 \end{pmatrix} = \begin{pmatrix} 2(-2x_1 + 2 + \lambda_1 x_1 + \lambda_2 x_1 - 2\lambda_2) \\ 2x_2(2 + \lambda_1 + \lambda_2) \end{pmatrix} = 0$$

   - if $x_2 \neq 0$:

     $$\begin{aligned} &\lambda_1 = -(2+\lambda_2) \\ &\Rightarrow -2x_1 + 2 - (2+\lambda_2)x_1 + \lambda_2 x_1 - 2\lambda_2 = -2(2x_1 - 1 + \lambda_2) = 0 \Rightarrow \lambda_2 = 1 - 2x_1 \end{aligned}$$

     - if $\lambda_1 = 0 \Rightarrow \lambda_2 = -2$, since $\lambda_2 < 0$ this is not a valid KKT point.
     - if $\lambda_1 \neq 0 \Rightarrow \lambda_1 > 0 \Rightarrow \lambda_2 < -2$ as $\lambda_2 < 0$ this is not a valid KKT point

     as for any $\lambda_1 \geq 0$ we have $\lambda_2 < 0$; we conclude that $x_2 \neq 0$ is an invalid solution.

- else if $x_2 = 0$:
  - if $\lambda_1 \neq 0 \Rightarrow x_1 = \pm 2$ and as per the second constraint $x_1 = -2$ is invalid. $\Rightarrow x_1 = 2 \Rightarrow \lambda_2 = 0 \Rightarrow \lambda_1 = 1$.
  - else if $\lambda_1 = 0$:
    if $\lambda_2 = 0 \Rightarrow x_1 = 1$.
    else if $\lambda_2 \neq 0 \Rightarrow x_1 = 1$ or $x_1 = 3$, however $x_1 = 3$ violates the first constraint $\Rightarrow x_1 = 1 \Rightarrow \lambda_2 = 0$. Which contradicts the assumption that $\lambda_2 \neq 0$.

  We conclude that the valid KKT points are $(1, 0)$ with $\lambda_1 = \lambda_2 = 0$ (a stationary point of the original objective function) and $(2, 0)$ with $\lambda_1 = 1, \lambda_2 = 0$.

3. As the objective function and constraints are continuously differentiable and the constraints define a compact set, then satisfying the KKT conditions is necessary for optimality. We will calculate the value of the objective function for both cases:
   $f(1, 0) = 2$,
   $f(2, 0) = 0$.
   As this is a minimization problem, we conclude that the optimal solution is $(2, 0)$.

Solution 2.

1. We require the following to define a non-empty set:

   $$f(x) = x^T Q x + 2b^T x + c \leq 0$$

   As Q is PD and $f(x)$ is a quadratic convex function, it attains a global minimum at the unique point at which $\nabla_x f(x) = 0 \Rightarrow x_{min} = -Q^{-1}b$. As this is the global minimum of f, any point $x \neq x_{min}$ satisfies $f(x) > f(x_{min})$ hence if $f(x_{min}) > 0$ then for all $x \in \mathbb{R}^n$ the constraint defines an empty set.

   $$\begin{aligned} &f(-Q^{-1}b) = (Q^{-1}b)^T Q Q^{-1} b - 2b^T Q^{-1} b + c = -b^T Q^{-1} b + c \leq 0 \\ &\Rightarrow ||Q^{-1/2} b||_2^2 \leq c \end{aligned}$$

   notice that as Q is PD $\Rightarrow Q^{-1}$ is PD $\Rightarrow Q^{-1/2}$ is PD $\Rightarrow c \geq 0$ (where if $c = 0$ then $b = 0$ by definition of a PD matrix).

2. The objective and constraint are continuously differentiable. In addition the constraint is convex $\Rightarrow$ following Slater's condition, we require that there exists some $x \in \mathbb{R}^n$ such that $f(x) < 0$. As we have shown above, the minimal value is attained at $x_{min} = -Q^{-1}b$ - for any Q, b, c such that $f(-Q^{-1}b) < 0$ KKT is a necessary condition.

3. The objective and constraint are continuously differentiable convex functions. Therefore, KKT is a sufficient condition for optimality as long as the constraint defines a non-empty set (conditions solved in sec. 1).

4. We now assume that $f(-Q^{-1}b) \leq 0$, the KKT conditions are

$$a + 2\lambda(Qx + b) = 0$$
$$x = -Q^{-1}(\frac{a}{2\lambda} + b)$$

as the objective is an affine function, the solution lies on the boundary of the set $\Rightarrow \lambda > 0$.

$$\begin{aligned}
&x^T Qx + 2b^T x + c = 0 \\
&(Q^{-1}(\frac{a}{2\lambda} + b))^T QQ^{-1}(\frac{a}{2\lambda} + b) - 2b^T Q^{-1}(\frac{a}{2\lambda} + b) + c = 0 \\
&\Rightarrow \frac{1}{4\lambda^2} a^T Q^{-1} a + b^T Q^{-1} b - \frac{1}{\lambda} a^T Q^{-1} b - 2b^T Q^{-1} b - \frac{1}{\lambda} a^T Q^{-1} b + c = 0 \\
&\Rightarrow \frac{1}{4\lambda^2} a^T Q^{-1} a - 2b^T Q^{-1} b + c = 0 \\
&\Rightarrow \lambda = \sqrt{\frac{a^T Q^{-1} a}{4(b^T Q^{-1} b - c)}}
\end{aligned}$$

substituting back into x:

$$x^* = -Q^{-1} \left(a\sqrt{\frac{b^T Q^{-1} b - c}{a^T Q^{-1} a}} + b \right)$$

Solution 3.

1. The Lagrangian of the problem is:

$$L(x, \lambda) = 2x_1^2 + (x_2 - 4)^2 + \lambda(-x_1^2 + 3kx_2)$$

Therefore, for the KKT conditions we require:

$$0 = \nabla L(x, \lambda) = \begin{pmatrix} (4 - 2\lambda)x_1 \\ 2(x_2 - 4) + 3k\lambda \end{pmatrix}$$
$$0 = \lambda(-x_1^2 + 3kx_2)$$

Case 1: $\lambda = 0 \Rightarrow x_1 = 0, x_2 = 4$: not a feasible point, since $k > 0$.
Case 2: $\lambda > 0, x_1 \neq 0$ and then:

$$\Rightarrow \lambda = 2 \Rightarrow x_2 = 4 - 3k \Rightarrow -2x_1^2 + 6k(4 - 3k) = 0 \Rightarrow x_1 = \pm\sqrt{3k(4 - 3k)}$$

which is feasible for $0 < k < \frac{4}{3}$.
Case 3: $\lambda > 0, x_1 = 0$ and then:

$$\begin{cases} 2x_2 - 8 + 3k\lambda = 0 \\ 3kx_2\lambda = 0 \end{cases}$$

which gives us $x_2 = 0, \lambda = \frac{8}{3k}$.
In conclusion, the only KKT points are $(\pm\sqrt{3k(4 - 3k)}, 4 - 3k)$ and $(0, 0)$.

2. The objective f and constraint f_1 are continuously differentiable functions. Second order optimality requires that $y^T \nabla^2_{xx} L(x^*, \lambda) y \geq 0$ for all $y \in \Delta(x^*)$ where $\Delta(x^*) = \{d \in \mathbb{R}^n : \nabla f_1(x^*)^T d = 0\}$. Since all the KKT points we found above are regular (there is only one constraint), it remains to check which of them obeys the second order necessary conditions above.

$$\nabla f_1(x) = \begin{pmatrix} -2x_1 \\ 3k \end{pmatrix}$$
$$\nabla^2_{xx} L(x, \lambda) = \begin{pmatrix} 4-2\lambda & 0 \\ 0 & 2 \end{pmatrix}$$

$(0,0)$:

$$\nabla f_1((0,0)) = (0, 3k)^T$$
$$\Delta((0,0)) = \{(z,0)^T : z \in \mathbb{R}\}$$
$$y^T \nabla^2_{xx} L\left((0,0), \frac{8}{3k}\right) y = (z,0) \begin{pmatrix} 4 - 2 \cdot \frac{8}{3k} & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} z \\ 0 \end{pmatrix} = \frac{12k-16}{3k} z^2 \geq 0, \forall k \geq \frac{4}{3}$$

The existence of second order conditions for $(0,0)$ means it is a local minimum (for any $k \geq \frac{4}{3}$. Since the other KKT points are not feasible in this range of k, we conclude $(0,0)$ is a global optimum for this range.

$(\pm\sqrt{3k(4-3k)}, 4-3k)$:

$$\nabla f_1((\pm\sqrt{3k(4-3k)}, 4-3k)) = (\mp 2\sqrt{3k(4-3k)}, 3k)^T$$
$$\Delta((\pm\sqrt{3k(4-3k)}, 4-3k)) = \{z \in \mathbb{R}^2 : \mp 2\sqrt{3k(4-3k)} \cdot z_1 + 3k z_2 = 0\} =$$
$$= \left\{ \left(z, \pm \frac{2\sqrt{3k(4-3k)}}{3k} z \right) : z \in \mathbb{R} \right\}$$
$$y^T \nabla^2_{xx} L\left((\pm\sqrt{3k(4-3k)}, 4-3k), 2\right) y = \left(z, \pm \frac{2\sqrt{3k(4-3k)}}{3k} z \right) \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} z \\ \pm \frac{2\sqrt{3k(4-3k)}}{3k} z \end{pmatrix} =$$
$$= 2 \frac{4 \cdot 3k(4-3k)}{9k^2} z^2 \geq 0, \forall 0 < k < \frac{4}{3}$$

which fits the feasibility limits of this point. Therefore, we conclude that for $0 < k < \frac{4}{3}$, there are two global optima, at $(\pm\sqrt{3k(4-3k)}, 4-3k)$.

Solution 4. Since x^* is a local minimum of the problem and since we can write each equality constraint as two inequality constraints, and also since f, g, s, h are continuously differentiable, the Fritz-John conditions are satisfied and we have $\tilde{\lambda}_0, \tilde{\lambda}_1, ..., \tilde{\lambda}_{m+p+2q} \geq 0$ (and not all zeros) such that:

$$
\begin{aligned}
\tilde{\lambda}_0 \nabla f(x^*) + \sum_{i=1}^{m} \tilde{\lambda}_i \nabla g_i(x^*) + \sum_{i=m+1}^{m+p} \tilde{\lambda}_i \nabla h_{i-m}(x^*) + \\
+ \sum_{i=m+p+1}^{m+p+q} \tilde{\lambda}_i \nabla s_{i-m-p}(x^*) - \sum_{i=m+p+q+1}^{m+p+2q} \tilde{\lambda}_i \nabla s_{i-m-p-q}(x^*) = 0 \\
\tilde{\lambda}_i g_i(x^*) = 0, i = 1, ..., m \\
\tilde{\lambda}_i h_{i-m}(x^*) = 0, i = m+1, ..., m+p \\
\tilde{\lambda}_i s_{i-m-p}(x^*) = 0, i = m+p+1, ..., m+p+q \\
\tilde{\lambda}_i s_{i-m-p-q}(x^*) = 0, i = m+p+q+1, ..., m+p+2q
\end{aligned}
$$

We'd like to set $\tilde{\lambda}_0 = 1$, which would then give us the existence of the other required multipliers λ, η, μ. This would be possible for any $\tilde{\lambda}_0 > 0$, since we would then be able to divide the whole equation by $\tilde{\lambda}_0$ and obtain the required result.
Assume $\tilde{\lambda}_0 = 0$. We now know there must exist $\tilde{\lambda}_1, ..., \tilde{\lambda}_{m+p+2q} \geq 0$ and not all zeros such that:

$$
\begin{aligned}
\sum_{i=1}^{m} \tilde{\lambda}_i \nabla g_i(x^*) + \sum_{i=m+1}^{m+p} \tilde{\lambda}_i \nabla h_{i-m}(x^*) + \sum_{i=m+p+1}^{m+p+q} \tilde{\lambda}_i \nabla s_{i-m-p}(x^*) - \\
- \sum_{i=m+p+q+1}^{m+p+2q} \tilde{\lambda}_i \nabla s_{i-m-p-q}(x^*) = 0 \qquad (1) \\
\tilde{\lambda}_i g_i(x^*) = 0, i = 1, ..., m \\
\tilde{\lambda}_i h_{i-m}(x^*) = 0, i = m+1, ..., m+p \\
\tilde{\lambda}_i s_{i-m-p}(x^*) = 0, i = m+p+1, ..., m+p+q \\
\tilde{\lambda}_i s_{i-m-p-q}(x^*) = 0, i = m+p+q+1, ..., m+p+2q
\end{aligned}
$$

By the gradient inequality and the generalized Slater's condition, we have:

$$
\begin{aligned}
&0 > g_i(\hat{x}) \geq g_i(x^*) + \nabla g_i(x^*)^T(\hat{x} - x^*), \;\; i = 1, ..., m \\
&0 \geq h_{j-m}(\hat{x}) = h_{j-m}(x^*) + \nabla h_{j-m}(x^*)^T(\hat{x} - x^*), \;\; j = m+1, ..., m+p \\
&0 = s_{k-m-p}(\hat{x}) = s_{k-m-p}(x^*) + \nabla s_{k-m-p}(x^*)^T(\hat{x} - x^*) \;\; k = m+p+1, ..., m+p+q \\
&0 = -s_{k-m-p-q}(\hat{x}) = -s_{k-m-p-q}(x^*) - \nabla s_{k-m-p-q}(x^*)^T(\hat{x} - x^*) \;\; k = m+p+q+1, ..., m+p+2q
\end{aligned}
$$

Multiplying each equation by the appropriate $\tilde{\lambda}_i$, we have:

$$
\begin{aligned}
&0 > \tilde{\lambda}_i g_i(x^*) + \tilde{\lambda}_i \nabla g_i(x^*)^T(\hat{x} - x^*) = \tilde{\lambda}_i \nabla g_i(x^*)^T(\hat{x} - x^*), \;\; i = 1, ..., m \\
&0 \geq \tilde{\lambda}_j h_{j-m}(x^*) + \tilde{\lambda}_j \nabla h_{j-m}(x^*)^T(\hat{x} - x^*) = \tilde{\lambda}_j \nabla h_{j-m}(x^*)^T(\hat{x} - x^*), \;\; j = m+1, ..., m+p \\
&0 = \tilde{\lambda}_k s_{k-m-p}(x^*) + \tilde{\lambda}_k \nabla s_{k-m-p}(x^*)^T(\hat{x} - x^*) = \tilde{\lambda}_k \nabla s_{k-m-p}(x^*)^T(\hat{x} - x^*), \;\; k = m+p+1, ..., m+p+q \\
&0 = -\tilde{\lambda}_k s_{k-m-p-q}(x^*) - \tilde{\lambda}_k \nabla s_{k-m-p-q}(x^*)^T(\hat{x} - x^*) = -\tilde{\lambda}_k \nabla s_{k-m-p-q}(x^*)^T(\hat{x} - x^*), \\
&\;\; k = m+p+q+1, ..., m+p+2q
\end{aligned}
$$

Now we define $A = (\tilde{\lambda}_1 \nabla g_1(x^*), ..., \tilde{\lambda}_m \nabla g_m(x^*))^T$ and:

$$\begin{aligned} B = (&\tilde{\lambda}_{m+1} \nabla h_1(x^*), ..., \tilde{\lambda}_{m+p} \nabla h_p(x^*), \\ &\tilde{\lambda}_{m+p+1} \nabla s_1(x^*), ..., \tilde{\lambda}_{m+p+q} \nabla s_q(x^*), \\ &-\tilde{\lambda}_{m+p+q+1} \nabla s_1(x^*), ..., -\tilde{\lambda}_{m+p+2q} \nabla s_q(x^*))^T \end{aligned}$$

Then, defining $d = (\hat{x} - x^*)$, using the above gradient inequalities we have:

$$\begin{aligned} Ad &< 0 \\ Bd &\leq 0 \end{aligned}$$

meaning d solves the system $Ad < 0, Bd \leq 0$. Using Motzkin's Lemma, this means that there do not exist $u, y \geq 0, u \neq 0$ such that:

$$A^T u + B^T y = 0$$

specifically, the equality does not hold for $u = (1, ..., 1)^T \in \mathbb{R}^m$ and $y = (1, ..., 1)^T \in \mathbb{R}^{p+2q}$, and therefore equality 1 does not hold.
This means we have $\tilde{\lambda}_0 > 0$ and therefore we can define:

$$\begin{aligned} &\forall i = 1, ..., m : && \lambda_i = \tfrac{\tilde{\lambda}_i}{\tilde{\lambda}_0} \\ &\forall j = m+1, ..., m+p : && \eta_{j-m} = \tfrac{\tilde{\lambda}_j}{\tilde{\lambda}_0} \\ &\forall k = m+p+1, ..., m+p+q : && \mu_{j-m-p} = \tfrac{\tilde{\lambda}_k - \tilde{\lambda}_{k+q}}{\tilde{\lambda}_0} \end{aligned}$$

which gives us $\lambda \geq 0$, $\eta \geq 0$ and $\mu \in \mathbb{R}^q$, as required.

Solution 5.

1. The claim we are trying to show is equivalent to $-b \in \text{Range}(A - \lambda_{min}(A)I) \Rightarrow q_1^T b = 0$. If $-b \in \text{Range}(A - \lambda_{min}(A)I)$, we can write $-b$ as $(A - \lambda_{min}(A)I)x$ for some $0 \neq x \in \mathbb{R}^n$. Therefore:

$$q_1^T b = -q_1^T (A - \lambda_{min}(A)I)x = (-q_1^T \lambda_1 + q_1^T \lambda_1 I)x = 0 \ \ \forall x \in \mathbb{R}^n$$

2. Applying the Newton method to ϕ_1 has a major drawback of infinite derivatives - when nearing the eigenvalues of A, we have $\phi_1'(\lambda) \to \pm\infty$, and therefore we would get "stuck" in those regions (the step size would be close to 0). On ϕ_2, on the other hand, no such derivatives exist and we can safely use the Netwon method.

3. First, we note since $\lambda > \lambda_{min}(A)$, $R^T R = A + \lambda^{(k)} I$ is invertible. Since R and R^T are diagonal matrices with positive values along the diagonal, they are also both separately

invertible. Therefore:

$$
\begin{aligned}
p^{(k)} &= -(R^TR)^{-1}b = -(A+\lambda^{(k)}I)^{-1}b \\
q^{(k)} &= R^{-T}p^{(k)} \\
\Rightarrow ||p^{(k)}|| &= ||(A+\lambda^{(k)}I)^{-1}b|| \\
\Rightarrow ||q^{(k)}||^2 &= ||R^{-T}p^{(k)}||^2 = (R^{-T}p^{(k)})^T(R^{-T}p^{(k)}) = p^{(k)T}R^{-1}R^{-T}p^{(k)} = \\
&= p^{(k)T}(R^TR)^{-1}p^{(k)} = p^{(k)T}(A+\lambda^{(k)}I)^{-1}p^{(k)} = \\
&= ((A+\lambda^{(k)}I)^{-1}b)^T(A+\lambda^{(k)}I)^{-1}(A+\lambda^{(k)}I)^{-1}b = \\
&= ((A+\lambda^{(k)}I)^{-\frac{3}{2}}b)^T((A+\lambda^{(k)}I)^{-\frac{3}{2}}b) = \\
&= ((Q\Lambda Q^T+\lambda^{(k)}IQQ^T)^{-\frac{3}{2}}b)^T((Q\Lambda Q^T+\lambda^{(k)}IQQ^T)^{-\frac{3}{2}}b) = \\
&= (Q^{-\frac{3}{2}T}(\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^{-\frac{3}{2}}b)^T(Q^{-\frac{3}{2}T}(\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^{-\frac{3}{2}}b) = \\
&= ((\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^{-\frac{3}{2}}b)^TQ^{-\frac{3}{2}}Q^{-\frac{3}{2}T}((\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^{-\frac{3}{2}}b) = \\
&= ((\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^{-\frac{1}{2}}Q^Tb)^TQ^{-\frac{1}{2}}Q^TQQ^{-\frac{1}{2}T}((\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^{-\frac{1}{2}}Q^Tb) = \\
&= ((\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^{-\frac{1}{2}}Q^Tb)^TQ^{-\frac{1}{2}}Q^{-\frac{1}{2}T}((\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^{-\frac{1}{2}}Q^Tb) = \\
&= ((\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^Tb)^T((\Lambda+\lambda^{(k)}I)^{-\frac{3}{3}}Q^Tb) = ||(\Lambda+\lambda^{(k)}I)^{-\frac{3}{2}}Q^Tb||^3
\end{aligned}
$$

For $\phi_2(\lambda) = \frac{1}{||x(\lambda)||} - \frac{1}{\alpha}$ (dropping the (k) notation for convenience) we have:

$$
\begin{aligned}
\phi_2'(\lambda) &= \frac{1}{||x(\lambda)||^2}\cdot\frac{x(\lambda)^T}{||x(\lambda)||}\cdot\frac{\partial}{\partial\lambda}x(\lambda) = \\
&= -\frac{1}{||(A+\lambda I)^{-1}b||^2}\cdot\frac{(-Q(\Lambda+\lambda I)^{-1}Q^Tb)^T}{||(A+\lambda I)^{-1}b||}\cdot\frac{\partial}{\partial\lambda}(-Q(\Lambda+\lambda I)^{-1}Q^Tb) = \\
&= \frac{1}{||(A+\lambda I)^{-1}b||^2}\cdot\frac{(Q(\Lambda+\lambda I)^{-1}Q^Tb)^T}{||(A+\lambda I)^{-1}b||}\cdot\left(-Q\frac{\partial}{\partial\lambda}((\Lambda+\lambda I)^{-1})Q^Tb\right) = \\
&= \frac{1}{||(A+\lambda I)^{-1}b||^2}\cdot\frac{(Q(\Lambda+\lambda I)^{-1}Q^Tb)^T}{||(A+\lambda I)^{-1}b||}\cdot Q(\Lambda+\lambda I)^{-1}(\Lambda+\lambda I)^{-1}Q^Tb = \\
&= \frac{1}{||(A+\lambda I)^{-1}b||^2}\cdot\frac{((\Lambda+\lambda I)^{-1}Q^Tb)^T}{||(A+\lambda I)^{-1}b||}\cdot(\Lambda+\lambda I)^{-1}(\Lambda+\lambda I)^{-1}Q^Tb = \\
&= \frac{1}{||p||^3}\cdot||(\Lambda+\lambda I)^{-\frac{3}{2}}Q^Tb||^2 = \frac{1}{||p||^3}\cdot||q||^2
\end{aligned}
$$

the Newton method step is:

$$\begin{aligned}\lambda^{(k+1)} &= \lambda^{(k)} - \frac{\phi_2(\lambda^{(k)})}{\phi_2'(\lambda^{(k)})} = \lambda^{(k)} - \frac{\frac{1}{||x(\lambda^{(k)})||} - \frac{1}{\alpha}}{\phi_2'(\lambda^{(k)})} \\ &= \lambda^{(k)} + \frac{||x(\lambda^{(k)})|| - \alpha}{\alpha} \cdot \frac{1}{\phi_2'(\lambda^{(k)})} \frac{1}{||x(\lambda^{(k)})||} = \\ &= \lambda^{(k)} + \frac{||p^{(k)}|| - \alpha}{\alpha} \cdot \frac{1}{\phi_2'(\lambda^{(k)})} \frac{1}{||p^{(k)}||} = \\ &= \lambda^{(k)} + \left(\frac{||p^{(k)}||}{||q^{(k)}||}\right)^2 \left(\frac{||p^{(k)}|| - \alpha}{\alpha}\right)\end{aligned}$$

4. The newton method solves the TRS problem with this generated data in 623 iterations, while the bisection method solves it in 28. The solution using both methods is the same, and it is given by:

$$x_{TRS} = \begin{pmatrix} 0.9539 \\ 0.2903 \\ -0.4669 \\ 0.8667 \\ -0.8903 \\ -0.3982 \\ 0.0727 \\ -0.5673 \\ 1.7499 \\ 1.6420 \end{pmatrix}$$

Below is the code for our TRS solver:

```
function [ x ] = TRS( A, b, alpha, method )
% TRS solves the Trust Region Subproblem

eps = 1e-7;
[~, p] = chol(A);
% define function
phi = @(lambda) 1 / norm((A + lambda * eye(size(A))) \ b) - 1 / alpha;

% define method function
if 'bisection' == method
    finder = @(lb, ub) bisection(phi, lb, ub, eps);
else
    finder = @(lb, ub) newton(phi, A, b, alpha, ub, lb, eps);
end

if 0 == p
    disp('PD') % A is PD
```

```
    x = - A \ b;
    if norm(x) <= alpha
       return
    else
         lb = 0;
    end
else
   disp('Not PD')% A is not PD
   lmin = eigs(A, 1, 'SR');
   lb = -lmin + eps; % lower bound as defined in tirgul
end

% keep increasing ub until phi(ub) < 0 such that phi(ub) * phi(lb) < 0
ub = max(lb, 1);
while phi(ub) < 0
   ub = 2*ub;
end

% run function
lambda = finder(lb, ub);
% extract x
x = - (A + lambda * eye(size(A))) \ b;
% enjoy
end
```

Additionally, below is code for our Newton Method solver:

```
function [ lambda ] = newton( f, A, b, alpha, lambda, lb, eps)
%INPUT
%================
%f..................... a scalar function
%A..................... a symmetric matrix
%b..................... a vector
%alpha................. alpha
%lambda................ initial value
%lb.................... lower bound for lambda
%eps................... tolerance parameter
%OUTPUT
%================
%lambda................ a root of the equation f(x)=0

iter = 0;

while abs(f(lambda)) > eps
    R = chol(A + lambda * eye(size(A)));
    p = - (R' * R) \ b;
```

```
    q = R' \ b;
    lambda = max(lambda + (norm(p) / norm(q))^2 * (norm(p) - alpha) / alp
    fprintf('iter_numer = %3d current_sol %2.6f \n',iter,lambda);
    iter = iter + 1;
end
```

For the Bisection method, we used the same code as supplied in class.
All code, including a main script for generating data and running with both methods, is attached to this submission.