Homework 4

Sahit Mandala

February 24, 2016

Problem 1

Consider the NLP as given with solution and unique Lagrange multipliers (by LICQ) x^*, λ^* and consider the unconstrainted problem indicated by the quadratic penalty function $P_\sigma(x) := f(x) + \frac{1}{2\sigma}\sum_{i=1}^m c_i^2(x)$. Since we are searching for a local minimizer for this function, we know that such a solution x_k^* must satisfy $\nabla P_\mu(x_k^*) = 0$. Some calculus shows that $\nabla P_\mu(x) = \nabla f(x) + \frac{1}{2\sigma}\sum_{i=1}^m \nabla c_i^2(x) = \nabla f(x) + \frac{1}{2\sigma}\sum_{i=1}^m 2c_i(x)\nabla c_i(x) = \nabla f(x) + \frac{1}{\sigma}\sum_{i=1}^m c_i(x)\nabla c_i(x) = \nabla f(x) + \frac{1}{\sigma}c(x)^T A(x)$.

Consider $h(x,\lambda,\sigma) = \begin{bmatrix}\nabla f(x) - A(x)^T\lambda \\ -c(x) - \sigma\lambda\end{bmatrix}$

Notice that for $x^*, \lambda^*, \sigma = 0$ that $h(x^*,\lambda^*,0) = \begin{bmatrix}\nabla f(x^*) - A(x^*)^T\lambda \\ -c(x^*)\end{bmatrix} = 0$ by the KKT conditions on the original NLP. We also know that $F(x,\lambda,\sigma) = \begin{bmatrix}\nabla f(x) - A(x)^T\lambda \\ -c(x) - \sigma\lambda\end{bmatrix}$ is continuously differentiable in some neighborhood of $[x^*;\lambda^*;0]$ since $\nabla f(x), c(x), \lambda$ are all continuously differentiable in some neighborhood of $[x^*;\lambda^*;0]$. We also see that $\nabla_{x,\lambda}h(x,\lambda,\sigma) = \begin{bmatrix}\nabla^2_{xx}L(x^*,\lambda^*) & -A(x)^T \\ -A(x) & -\sigma I\end{bmatrix}$, noting that $\nabla^2_{xx}L(x^*,\lambda^*) = \nabla^2 f(x^*) - \nabla A(x^*)^T\lambda$. We know that $\nabla_{x,\lambda}h(x,\lambda,0) = \begin{bmatrix}\nabla^2_{xx}L(x^*,\lambda^*) & -A(x^* \\ -A(x^*) & 0\end{bmatrix}$ is nonsingular since assumption 18.2 holds at x^* with Lagrange multipliers λ^* (assumption 18.2 follows from LICQ and the second order conditions) from which it follows that it is nonsingular.

By the implicit function theorem, we see that we can choose some open set N_x around $[x^*;\lambda^*]$, N_σ around $\sigma = 0$, and some function continuous $z : N_\sigma \to N_x$ s.t. $h(z(t),t) = 0$ for all $t \in N_\mu$ (we can say that $z(\sigma) = [z_x(\sigma); z_\lambda(\sigma)]$). For these $\sigma \in N_\sigma$, we see that $h(z(\sigma),\sigma) = \begin{bmatrix}\nabla f(z_x(\sigma)) - A(z_x(\sigma))^T z_\lambda(\sigma) \\ -c(z_x(\sigma)) - \sigma z_\lambda(\sigma)\end{bmatrix} = 0$. Note that this implies that $-c(z_x(\sigma)) - \sigma z_\lambda(\sigma) = 0$, or equivalently, $-\frac{1}{\sigma}c(z_x(\sigma)) = z_\lambda(\sigma)$. So for a given σ, the point $z_x(\sigma), z_\lambda(\sigma)$ satisfies $\nabla P_\sigma = \nabla f(z_x(\sigma)) + \frac{1}{\sigma}c(z_x(\sigma))^T A(z_x(\sigma)) = \nabla f(z_x(\sigma)) + (-z_\lambda(\sigma))^T A(z_x(\sigma)) = 0$. We also see that $h(z(0),0) = 0$, which is true only when $z(0) = [x^*;\lambda^*]$, so by the continuity of z, as $\sigma \to 0$, $z_x(\sigma) \to x^*, z_\lambda(\sigma) \to \lambda^*$. Hence, we see that for sufficiently small $\sigma > 0$, a minimizer for P_σ will be within the neighborhood of x^*.

To show that this point in a local minimizer, we must show the next necessary condition that $\nabla^2 P_\mu(x)$ is positive semidefinite at this point. Notice that (using $x = z_x(\sigma), \lambda = z_\lambda(\sigma) = -\frac{1}{\sigma}c(x)$:

$\nabla^2 P_\mu(x) = \nabla^2 f(x) + \frac{1}{\sigma}\sum_{i=1}^m \nabla c_i(x)^T\nabla c_i(x) + \frac{1}{\sigma}\sum_{i=1}^m c_i(x)\nabla^2 c_i(x) = \nabla^2_{xx}L_\sigma(x,\lambda) + \frac{1}{\sigma}A(x)^T A(x)$.

Let some y be given and consider $y^T\nabla^2 P_\mu(x)y = y^T\nabla^2_{xx}L_\sigma(x,\lambda)y + \frac{1}{\sigma}y^T A(x)^T A(x)y$. Notice that as $\sigma \to 0$, $\nabla^2_{xx}L_\sigma(x,\lambda) \to \nabla^2_{xx}L_\sigma(x^*,\lambda^*)$ since $x = z_x(\sigma) \to x^*, \lambda = z_\lambda(\sigma) = -\frac{1}{\sigma}c(x) \to \lambda^*$. We know that $\nabla^2_{xx}L_\sigma(x^*,\lambda^*)$ has strictly positive eigenvalues by the 2nd order conditions, so we can set $\epsilon > 0$ as the smallest eigenvalue of $\nabla^2_{xx}L_\sigma(x^*,\lambda^*)$. Since convergent matricies are convergent in their eigenvalues, for sufficiently small σ, we can make $\nabla^2_{xx}L_\sigma(x,\lambda)$ s.t. the each eigenvalue of $\nabla^2_{xx}L_\sigma(x,\lambda)$ is within $\frac{\epsilon}{2}$ of some eigenvalue of $\nabla^2_{xx}L_\sigma(x^*,\lambda^*)$, and since ϵ is the smallest of all of those, this guarantees that $\nabla^2_{xx}L_\sigma(x,\lambda)$ is positive semidefinite. Next, consider $y^T\frac{1}{\sigma}A(x)^T A(x)y$. Note that because $A(x)^T A(x)$ is symmetric, it readily follows that it has positive eigenvalues and thus is positive semidefinite. So then $\frac{1}{\sigma}A(x)^T A(x)$ is clearly also positive semidefinite for $\sigma > 0$. So then the sum of these 2 positive semidefinite matricies is also positive semidefinite, $\nabla^2 P_\mu(x)$. Hence, we have shown that $\nabla^2 P_\mu(x)$ is positive semidefinite for sufficiently small σ, and coupled with the above results showing

$\nabla P_\mu(z_x(\sigma)) = 0$ as well as $\sigma \to 0$, $z_x(\sigma) \to x^*, z_\lambda(\sigma) \to \lambda^*$, we have shown that that these points are local minimizers and there exists such a local minimizer in a neighborhood of x^* for a given σ small enough.

Problem 2

In our current iteration k with given x^*, λ^*, the Lagrangian can be written as $\nabla L_{xx}(x^*, \lambda^*) = \nabla f_k - A_k^T \lambda^*$. If we substitute this into the original objective function $(f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla L_{xx}(x^*, \lambda^*) p)$ for $\nabla f_k^T p$, we see that $f_k + \nabla L_{xx}(x^*, \lambda^*)^T p + \frac{1}{2} p^T \nabla L_{xx}(x^*, \lambda^*) p = f_k + (\nabla f_k - A_k^T \lambda^*)^T p + \frac{1}{2} p^T \nabla L_{xx}(x^*, \lambda^*) p =$
$f_k + \nabla f_k^T p - (A_k^T \lambda^*)^T p + \frac{1}{2} p^T \nabla L_{xx}(x^*, \lambda^*) p$
Notice that $-(A_k^T \lambda^*)^T p = -(\lambda^*)^T (A_k^T)^T p = -(\lambda^*)^T A_k p$
But we know that $A_k p + c_k = 0$ from the first constraint in the problem, so $A_k p = -c_k$ and thus $-(\lambda^*)^T A_k p = -(\lambda^*)^T - c_k = (\lambda^*)^T c_k$. But this is merely a constant (say, $r = (\lambda^*)^T c_k$) since λ^*, c_k are both given. So $f_k + \nabla f_k^T p - (A_k^T \lambda^*)^T p + \frac{1}{2} p^T \nabla^2 L_{xx}(x^*, \lambda^*) p = f_k + \nabla f_k^T p + r + \frac{1}{2} p^T \nabla^2 L_{xx}(x^*, \lambda^*) p$
But the problem of minimizing this objective is the same as minimizing this objective without the constant, so min $f_k + \nabla f_k^T p + r + \frac{1}{2} p^T \nabla^2 L_{xx}(x^*, \lambda^*) p \cong$ min $f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla^2 L_{xx}(x^*, \lambda^*) p = f_k + \nabla L_{xx}(x^*, \lambda^*)^T p + \frac{1}{2} p^T \nabla^2 L_{xx}(x^*, \lambda^*) p$. Hence, we have shown that the original problem with $\nabla f_k^T p$ is equivalent to the problem with $f_k + \nabla L_{xx}(x^*, \lambda^*)^T p$.

Problem 3

First, we shall notice that algorithm 18.1 is structure based upon algorithm 11.1. We consider $x'_k = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix}$ as our variable with $x'_0 = \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix}$. We note that given the formulation of l_k and λ_k, we see that $\lambda_{k+1} = \lambda_k + p_\lambda$ from the formulation in 18.6.
Then the body of the loop in algorithm 18.1 amounts to solving a linear system of the form:
$\begin{bmatrix} \nabla^2_{xx} L_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}$
But this is equivalent to solving the following problem with the substitution for $\lambda_{k+1} = \lambda_k + p_\lambda$
$\begin{bmatrix} \nabla^2_{xx} L_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = J(x'_k) \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = r(x'_k) = \begin{bmatrix} -\nabla f_k + A_k^T \lambda_k \\ -c_k \end{bmatrix}$
After solving this linear equation for a solution $\begin{bmatrix} p_k \\ p_\lambda \end{bmatrix}$, we see that $x_{k+1} \leftarrow x_k + p_k$ and $\lambda_{k+1} \leftarrow \lambda_k + p_\lambda$ for the variable $x'_k = \begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix}$, after which we repeat the process. This formulates the algorithm equivalently to Newton's algorithm in 11.1.
We next note that because assumption 18.2 holds, we know that the Jacobian $A(x)$ has full row rank and that $\nabla^2_{xx} L_k$ is positive definite on the tangent space of constraints. This implies that the matrix $J(x'_k)$ above is nonsingular. We also know that the solution x^* to the original NLP has some unique Lagrange multipliers λ^* (by LICQ) satisfying $r(\begin{bmatrix} x^* \\ \lambda^* \end{bmatrix}) = 0$ since $r(\begin{bmatrix} x^* \\ \lambda^* \end{bmatrix}) = \begin{bmatrix} -\nabla f_k(x^*) + A_k^T \lambda \\ -c_k(x^*) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ by the KKT conditions and constraints. So if $\{x'_k\}$ are within some neighborhood of $\begin{bmatrix} x^* \\ \lambda^* \end{bmatrix}$, our algorithm will converge to some solution in some Q-superlinear time by theorem 11.2.
Because f, c are twice Lipchitz differentiable near x^*, then $\nabla f, \nabla c_i$ are Lipshitz continuously differentiable. So then $A_k^T \lambda_k$ is also Lipshitz continuously differentiable (product and sum of Lipshitz continuously differentiable functions). So overall $r(x'_k) = \begin{bmatrix} -\nabla f_k + A_k^T \lambda_k \\ -c_k \end{bmatrix}$ is also Lipshitz continuously differentiable (on x and λ) as the sums and differences of these terms.
This implies by theorem 11.2 that our algorithm has quadratic convergence if $\{x'_k\}$ are within some neighborhood of $\begin{bmatrix} x^* \\ \lambda^* \end{bmatrix}$.

Problem 4

Program code:

```
% Homework 4
% Problem 4
answer = [-1.71, 1.59, 1.82, -0.763, -0.763]'

%Variables
x = sym('x',[5,1])
l = sym('l',[3,1]) %num constraints

%Initial choices
x_initial = [-1.8, 1.7, 1.9, -0.8, -0.8]'
l_initial = [1,-1,1]'

f = exp(x(1,1)*x(2,1)*x(3,1)*x(4,1)*x(5,1))-(0.5)*(x(1,1)^3+x(2,1)^3+1)^2
c1 = x(1,1)^2+x(2,1)^2+x(3,1)^2+x(4,1)^2+x(5,1)^2-10
c2 = x(2,1)*x(3,1)-5*x(4,1)*x(5,1)
c3 = x(1,1)^3+x(2,1)^3+1
c = [c1;c2;c3]

%Algo 18.1
%Find symbolic components
grad_f = gradient(f)
A = jacobian(c,x)
F = [gradient(f,x)-A'*l; c];
Fj = jacobian(F,[x;l])

%Setup
xk = x_initial;
lk = l_initial;
old_sol= [1;2;3;4;5;6;7;8]
sol = [8;7;6;5;4;3;2;1]

%No convergence test specified, chose a simple norm heuristic
while norm(sol-old_sol) > 1*10^-8
  old_sol = sol

  Fk = double(subs(Fj,[x;l],[xk;lk]));
  grad_fk = double(subs(grad_f,x,xk));
  ck = double(subs(c,x,xk));
  sol = inv(Fk)*-[grad_fk; ck] %SOlve linear eqn
  xk = xk + sol(1:5,1);
  lk = sol(6:8,1);
  i=i+1;
end

% Solution
```

xk

Problem 5

Consider $c(x) = x_1^2 + x_2^2 - 1$. For the given constraint we see that the Jacobian is:

$A_k = A(x_k) = [\nabla c(x)]^T = \begin{bmatrix} 2x_1 & 2x_2 \end{bmatrix}$, $c_k = c(x_k) = x_1^2 + x_2^2 - 1$

So we can evaluate this at the following points:

$(0,0)^T$: $A_k = \begin{bmatrix} 0 & 0 \end{bmatrix}, c_k = 0 + 0 - 1 = -1 \rightarrow A_k p + c_k = \begin{bmatrix} 0 & 0 \end{bmatrix} p - 1 = -1 = 0$

$(0,1)^T$: $A_k = \begin{bmatrix} 0 & 2 \end{bmatrix}, c_k = 0 + 2^2 - 1 = 3 \rightarrow A_k p + c_k = \begin{bmatrix} 0 & 2 \end{bmatrix} p + 3 = 2p_2 + 3 = 0$

$(0.1, 0.02)^T$: $A_k = \begin{bmatrix} 0.2 & 0.04 \end{bmatrix}, c_k = 0.1^2 + 0.02^2 - 1 = 0.0104 - 1 = -0.9896 \rightarrow$

$A_k p + c_k = \begin{bmatrix} 0.2 & 0.04 \end{bmatrix} p - 0.9896 = 0.2p_1 + 0.04p_2 - 0.9896 = 0$

$(-0.1, 0.02)^T$: $A_k = \begin{bmatrix} -0.2 & -0.04 \end{bmatrix}, c_k = (-0.1)^2 + (-0.02)^2 - 1 = 0.0104 - 1 = -0.9896 \rightarrow$

$A_k p + c_k = \begin{bmatrix} -0.2 & -0.04 \end{bmatrix} p - 0.9896 = 0.2p_1 - 0.04p_2 - 0.9896 = 0$