



SF2822 Applied nonlinear optimization, final exam
Thursday May 31 2018 8.00–13.00

Examiner: Anders Forsgren, tel. 08-790 71 27.

Allowed tools: Pen/pencil, ruler and eraser.

Note! Calculator is not allowed.

Solution methods: Unless otherwise stated in the text, the problems should be solved by systematic methods, which do not become unrealistic for large problems. Motivate your conclusions carefully. If you use methods other than what have been taught in the course, you must explain thoroughly.

Note! Personal number must be written on the title page. Write only one question per sheet. Number the pages and write your name on each page.

22 points are sufficient for a passing grade. For 20-21 points, a completion to a passing grade may be made within three weeks from the date when the results of the exam are announced.

1. Consider a particular nonlinear programming problem

$$\begin{aligned} & \text{minimize} && f(x) \\ (NLP) \quad & \text{subject to} && g_i(x) \geq 0, \quad i = 1, \dots, 24, \\ & && x \in \mathbb{R}^9, \end{aligned}$$

where $f : \mathbb{R}^9 \rightarrow \mathbb{R}$ and $g : \mathbb{R}^9 \rightarrow \mathbb{R}^{24}$ are twice continuously differentiable.

A highly reliable friend of yours named AR has written a Matlab file “`prob`” with syntax “`[f,gradf,g,A,HessL] = prob(x,lambda)`”. For given “`x`” and “`lambda`” (x and λ), the function returns

$$\begin{aligned} \mathbf{f} & f(x), \\ \mathbf{gradf} & \nabla f(x), \\ \mathbf{g} & g(x), \\ \mathbf{A} & A(x), \quad \text{i.e.,} \quad \begin{pmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_{24}(x)^T \end{pmatrix}, \\ \mathbf{HessL} & \nabla_{xx}^2 \mathcal{L}(x, \lambda), \quad \text{i.e.,} \quad \nabla^2 f(x) - \sum_{i=1}^{24} \lambda_i \nabla^2 g_i(x), \end{aligned}$$

for this particular problem (*NLP*). You do not have access to “`prob`”, but we assume that the function is correct.

A not so reliable friend of yours named AF claims that he has solved the problem by making use of “`prob`”, finding a solution x^* and corresponding Lagrange multiplier vector λ^* . It is not clear what AF means by “solved”. You have access to a printout of some calculations that he has done in Matlab to verify the solution “`xstar`” and corresponding Lagrange multiplier vector “`lambdastar`”. Since these calculations only involve Matlab built-in functions and “`prob`”, we assume that they are correct in spite of AF not being so reliable. The Matlab printout can be found at the end of the exam.

AF has made a number of claims. Based on the Matlab printout, comment each of AF’s claims. Motivate each of your comments.

- (a) Claim by AF: “The point x^* is a regular point to (*NLP*).”(2p)

- (b) Claim by AF: “The point x^* satisfies, together with λ^* , the first-order necessary optimality conditions to (NLP).” (2p)
- (c) Claim by AF: “The point x^* satisfies, together with λ^* , the second-order necessary optimality conditions to (NLP).” (2p)
- (d) Claim by AF: “The point x^* satisfies, together with λ^* , the second-order sufficient optimality conditions to (NLP).” (2p)
- (e) Claim by AF: “The point x^* is a global minimizer to (NLP).” (2p)

2. Consider the quadratic program (QP) defined by

$$\begin{aligned}
 & \text{minimize} && \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 \\
 \text{(QP)} & \text{subject to} && x_1 + x_2 \geq 6, \\
 & && x_1 \geq 0, \\
 & && x_2 \geq 2.
 \end{aligned}$$

Solve (QP) by an active-set method, with the initial point $x^{(0)}$ given by $x^{(0)} = (15 \ 3)^T$ with no constraints in the working set. The equality-constrained quadratic programs that arise need not be solved in a systematic way. They may for example be solved graphically. However, the values of the generated iterates $x^{(k)}$ and corresponding Lagrange multipliers $\lambda^{(k)}$ should be calculated. (10p)

3. Derive the expression for the symmetric rank-1 update, C_k , in a quasi-Newton update $B_{k+1} = B_k + C_k$ (10p)

4. Consider the nonlinear program (NLP) given by

$$\begin{aligned}
 & \text{minimize} && \frac{1}{2}x_1^2 + x_2 \\
 \text{(NLP)} & \text{subject to} && -x_1 - x_2^2 + 1 \geq 0, \\
 & && x \in \mathbb{R}^2.
 \end{aligned}$$

Assume that we want to solve (NLP) using a primal-dual interior method. Also assume that we initially choose $x^{(0)} = (2 \ 1)^T$, $\lambda^{(0)} = 2$, and $\mu = 1$.

- (a) When the constraints are in the form $g(x) \geq 0$, one may introduce slack variables s and rewrite the constraints as $g(x) - s = 0$, $s \geq 0$, when applying the interior method. Explain why the constraint $-x_1 - x_2^2 + 1 \geq 0$ needs to be rewritten in (NLP) for the given initial value $x^{(0)}$ (2p)
- (b) Formulate the system of linear equations to be solved in the first iteration of the primal-dual interior point method where the constraint $g(x) \geq 0$ is rewritten as $g(x) - s = 0$, $s \geq 0$. Use the given $x^{(0)}$ and μ . Select an appropriate value of $s^{(0)}$. Formulate the general form and then introduce explicit numerical values into the system of linear equations. (6p)
- (c) Assume that you have solved the system of linear equations that was formulated in Question 4b, so that you have obtained numerical values of its solution $\Delta x^{(0)}$, $\Delta s^{(0)}$, $\Delta \lambda^{(0)}$. Explain how you would choose $x^{(1)}$, $s^{(1)}$ and $\lambda^{(1)}$ (2p)

5. Consider the nonlinear optimization problem (*NLP*) given by

$$\begin{aligned}
 & \text{minimize} && -x_2 \\
 (\text{NLP}) & \text{subject to} && 1 - (x_1 - 1)^2 - x_2^2 \geq 0, \\
 & && 1 - (x_1 + 1)^2 - x_2^2 \geq 0.
 \end{aligned}$$

(a) We want to solve (*NLP*) by sequential quadratic programming. Let $x^{(0)} = (0 \ 1)^T$ and let $\lambda^{(0)} = (1 \ 1)^T$. Perform one iteration, i.e., calculate $x^{(1)}$ and $\lambda^{(1)}$. You may solve the subproblem in an arbitrary way that need not be systematic, and you do not need to perform any linesearch. (5p)

Hint: The following quantities, which you may use without verification, may be helpful,

$$\begin{aligned}
 \nabla_{xx}^2 \mathcal{L}(x^{(0)}, \lambda^{(0)}) &= \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}, & \nabla f(x^{(0)}) &= \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \\
 A(x^{(0)}) &= \begin{pmatrix} 2 & -2 \\ -2 & -2 \end{pmatrix}, & g(x^{(0)}) &= \begin{pmatrix} -1 \\ -1 \end{pmatrix},
 \end{aligned}$$

where the usual notation is used. In particular, the sign of λ is chosen such that $\mathcal{L}(x, \lambda) = f(x) - \lambda^T g(x)$.

(b) Problem (*NLP*) is a convex problem with global minimizer $x^* = (0 \ 0)$. However, x^* is not a regular point and no Lagrange multiplier vector exists. A consequence is that the multipliers of the SQP subproblem associated with (*NLP*) would go to infinity as the iterations proceed.

In order to create regularity and bound the multipliers, one may consider a reformulation of (*NLP*) according to

$$\begin{aligned}
 & \text{minimize} && f(x) + Me^T u \\
 (\text{NLP}') & \text{subject to} && g(x) + u \geq 0, \\
 & && u \geq 0,
 \end{aligned}$$

where M is a large positive (fixed) number, $e = (1 \ 1)^T$ and u are so-called elastic variables. Denote the Lagrange multipliers of the constraints $g(x) + u \geq 0$ by λ and denote the Lagrange multipliers of the onstraints $u \geq 0$ by η .

Assume that we want to solve (*NLP'*) by sequential quadratic programming. Formulate the QP subproblem (*QP'*) for given $x^{(k)}$, $u^{(k)}$, $\lambda^{(k)}$ and $\eta^{(k)}$. You need not give the explicit formulas for $f(x)$ and $g(x)$, instead use the notation $\nabla_{xx}^2 \mathcal{L}(x^{(k)}, \lambda^{(k)})$, $\nabla f(x^{(k)})$, $A(x^{(k)})$ and $g(x^{(k)})$ in your formulation. Let p and q denote the variables, corresponding to change in x and u respectively. Accordingly, let $p^{(k)}$, $q^{(k)}$ denote the optimal solution to (*SQP'*) and let $\lambda^{(k+1)}$, $\eta^{(k+1)}$ denote the corresponding Lagrange multipliers.

In addition, assume that $u^{(k)} = 0$ and $M > \max\{\lambda^{(k+1)}, \eta^{(k+1)}\}$. Show that in this situation it holds that $q^{(k)} = 0$ in addition to $p^{(k)}$ and $\lambda^{(k+1)}$ being optimal solution and Lagrange multipliers respectively of the corresponding QP subproblem associated with (*NLP*). (5p)

Good luck!

Matlab printout for Question 1.

```
>> xstar
```

```
xstar =
```

```
0.2500  
0.7500  
1.0000  
0.7500  
0.2500  
0.4330  
0.4330  
-0.4330  
-0.4330
```

```
>> lambdastar
```

```
lambdastar =
```

```
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0.4330  
0  
0  
0  
0  
0.2165  
0  
0  
0  
0.2165  
0  
0  
0
```

```
>> [f,gradf,g,A,HessL] = prob(xstar,lambdastar);
```

(Matlab printout continues on next page.)

```
>> [g lambdastar]

ans =

    0.2500    0
    0.4330    0
    0.5000    0
    0.4330    0
    0.2500    0
    0.2500    0
    0.4330    0
    0.5000    0
    0.4330    0
    0.7500    0
    0.2500    0
     0    0.4330
    0.2500    0
    0.7500    0
    0.7500    0
    0.2500    0
     0    0.2165
    0.2500    0
    0.7500    0
    0.2500    0
     0    0.2165
    0.7500    0
    0.2500    0
    0.7500    0

>> norm(gradf-A'*lambdastar)

ans =

    8.7771e-17

>> eps

ans =

    2.2204e-16

>> active=find(g<sqrt(eps))

active =

    12
    17
    21
```

(Matlab printout continues on next page.)

```
>> rank(A(active,:))

ans =

     3

>> Z=null(A(active,:))

Z =

    0.2222    -0.2750    0.3849    0.4763   -0.3849   -0.4763
    0.2750    0.2222    0.4763   -0.3849   -0.4763    0.3849
         0         0         0         0         0         0
    0.8889   -0.0393   -0.1925    0.0680    0.1925   -0.0680
    0.0393    0.8889    0.0680    0.1925   -0.0680   -0.1925
   -0.1925   -0.0680    0.6667    0.1179    0.3333   -0.1179
   -0.0680    0.1925   -0.1179    0.6667    0.1179    0.3333
    0.1925    0.0680    0.3333   -0.1179    0.6667    0.1179
    0.0680   -0.1925    0.1179    0.3333   -0.1179    0.6667

>> eig(Z'*HessL*Z)

ans =

    1.2990
    0.4330
   -0.5000
   -0.5000
    0.5000
    0.5000
```

(End of Matlab printout.)