Norwegian University of Science and Technology
Department of Mathematical Sciences

TMA4180 Optimization Theory
Spring 2013

Exercise set 6

This set of problems requires access to **Matlab/Matlab Optimization Toolbox**. The problems are taken from [1]. The solutions should be obvious, at least after you have seen the numerical solutions.

1 Find the minimum of *Wood's function*

$$\begin{aligned} f(x) &= 100(x_2 - x_1^2)^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 \\ &\quad + 10.1\left[(x_2-1)^2 + (x_4-1)^2\right] + 19.8(1-x_2)(1-x_4), \\ &x \in \mathbb{R}^4 \end{aligned}$$

using both `fminsearch` and `fminunc`. Set '`Display`' to '`iter`'.

Suggested starting point: $x_0 = [-3\ \ -1\ \ -3\ \ -1]'$, where $f(x_0) = 19192$.

Solution: The minimum is $x^* = (1,1,1,1)^{\mathrm{T}}$. *Some code:*

```
function F = woods(x)
% Wood's function
x1 = x(1); x2 = x(2); x3 = x(3); x4 = x(4);
F = 100*(x2-x1^2)^2 + (1-x1)^2 + 90*(x4-x3^2)^2 + (1-x3)^2 ...
    + 10.1*((x2-1)^2 + (x4-1)^2) + 19.8*(1-x2)*(1-x4);

options = optimset('Display', 'iter');
x0 = [-3, -1, -3, -1];
[x, fval] = fminunc('woods', x0, options)
[x, fval] = fminsearch('woods', x0)
```

2 Find the minimum of *Bigg's function*

$$\begin{aligned} f(x) &= \sum_{i=1}^{10} h_i(x)^2 = \sum_{i=1}^{10} \left\{\exp(-x_1 z_i) - x_3 \exp(-x_2 z_i) - y_i\right\}^2, \\ y_i &= \exp(-z_i) - 5\exp(-10 z_i), \\ z_i &= 0.1 \times i,\ \ i = 1, \cdots, 10,\ \ x \in \mathbb{R}^3 \end{aligned}$$

using the Least Square algorithm `lsqnonlin`. Write a function that computes both $h(x)$ and and $J(x)$ (Remember to set '`Jacobian`' to '`on`').

Suggested start value: $x_0 = [1\ 2\ 1]'$, where $f(x_0) = 1.55347...$

Warning: We have experienced some problems with older version of the routine.
Solution: The solution is $x^* = (1, 10, 5)^{\mathrm{T}}$. *Some code:*

```
function [F, J] = biggs(x)
% Bigg's function
z = 0.1*(1:100);
y = exp(-z) - 5*exp(-10*z);
F = (exp(-x(1)*z) - x(3)*exp(-x(2)*z) - y)';
if nargout > 1 % two output arguments
    J1 = -z.*exp(-x(1)*z);
    J2 = x(3)*z.*exp(-x(2)*z);
    J3 = -exp(-x(2)*z);
    J = [J1', J2', J3'];
end

options = optimset('Display', 'iter', 'Jacobian', 'on', 'TolX', 1.e-16);
x0 = [1, 2, 1]';
[x, resnorm] = lsqnonlin('biggs', x0, [], [], options)
```

3 Find the minimum of

$$f(x) = x'Ax - 2x_1,$$

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

$$x \in \mathbb{R}^{20}.$$

Start at $x = 0$ (In the objective function, x_1 denotes the first component of x).

Solution: Here we get $x^* = (20, 19, 18, \ldots, 1)^{\mathrm{T}}$.

Some code:

```
 %Build matrix A
A = zeros(20);
A(1,1) = 1;
A(2:20+1:end) = -1;
A(21:20+1:end) = -1;
A(22:20+1:end) = 2;

f = @(x) x'*A*x  - 2*x(1);

%Run optimization
options = optimset('Display', 'iter');
x0 = zeros(20,1);

[x, fval] = fminunc(f, x0, options)
```

References

[1] K. Schittkovski: *More Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Math. Systems No. 282, Springer 1987.