Norwegian University of Science and Technology
Department of Mathematical Sciences

TMA4180 Optimization Theory
Spring 2013

Exercise set 5

1 N&W Exercise 5.4 (p. 133 in 2nd edition). What important condition on the p-s is missing in the text? (*Hint*: Note that you may write $x_0 + P\sigma$, where $P = (p_0, p_1, \dots, p_{k-1})$ and $\sigma = (\sigma_0, \dots, \sigma_{k-1})^{\mathrm{T}}$.)

Solution:

In N&W Problem 5.4 (p. 133) we are going to show that if $f(x)$ is a strictly convex, quadratic function, then $h\colon \mathbb{R}^k \to \mathbb{R}$ defined by $h(\sigma) = f(x_0 + P\sigma)$ is also a quadratic and strictly convex function. We know that f is of the form $f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x + a$, where $\nabla^2 f = A > 0$.

We introduce $x_0 + P\sigma$ in the expression for f:

$$\begin{aligned}
h(\sigma) &= f(x_0 + \sigma_0 p_0 + \cdots + \sigma_{k-1}p_{k-1}) \\
&= f(x_0 + P\sigma) \\
&= \tfrac{1}{2}(x_0 + P\sigma)^{\mathrm{T}}A(x_0 + P\sigma) - b^{\mathrm{T}}(x_0 + P\sigma) + a \\
&= \tfrac{1}{2}(x_0^{\mathrm{T}}Ax_0 + \sigma^{\mathrm{T}}P^{\mathrm{T}}Ax_0 + x_0^{\mathrm{T}}AP\sigma + \sigma^{\mathrm{T}}P^{\mathrm{T}}AP\sigma) - b^{\mathrm{T}}(x_0 + P\sigma) + a \\
&= \tfrac{1}{2}\sigma^{\mathrm{T}}P^{\mathrm{T}}AP\sigma + (P^{\mathrm{T}}Ax_0 - P^{\mathrm{T}}b)^{\mathrm{T}}\sigma + a - b^{\mathrm{T}}x_0 + \tfrac{1}{2}x_0^{\mathrm{T}}Ax_0.
\end{aligned}$$

This is a quadratic function in σ. Since $A > 0$, $\sigma^{\mathrm{T}}P^{\mathrm{T}}AP\sigma > 0$ if and only if $P\sigma \neq 0$. Thus, $P^{\mathrm{T}}AP$ is positive definite (and hence h strictly convex) if and only if P has rank k. *The missing condition in the problem is that $\{p_k\}$ should be* linearly independent. *It is probable that $\{p_k\}$ were meant to be A-orthogonal, which in turn implies linear independence.*

2 In this problem we shall look at some statements you find in textbooks about the CG method.

The following simple MATLAB code for the CG method of a quadratic problem is also stated in the note on the Web:

```
ndim = 100; R = randn(ndim);
npot = .1;
A = (R'*R)^npot;
kappa = max(eig(A))/min(eig(A));
xsol = rand(ndim,1); b = A*xsol;
Norm2 = sqrt(xsol'*xsol); NormA = sqrt(xsol'*A*xsol);
x = zeros(size(b)); g = A*x-b; p = -g;
for loop = 1:ndim
    Ap = A*p;
    alfa = -(p'*g)./(p'*Ap);
    x = x + alfa*p;
    g = g + alfa*Ap; % g = A*x-b;
```

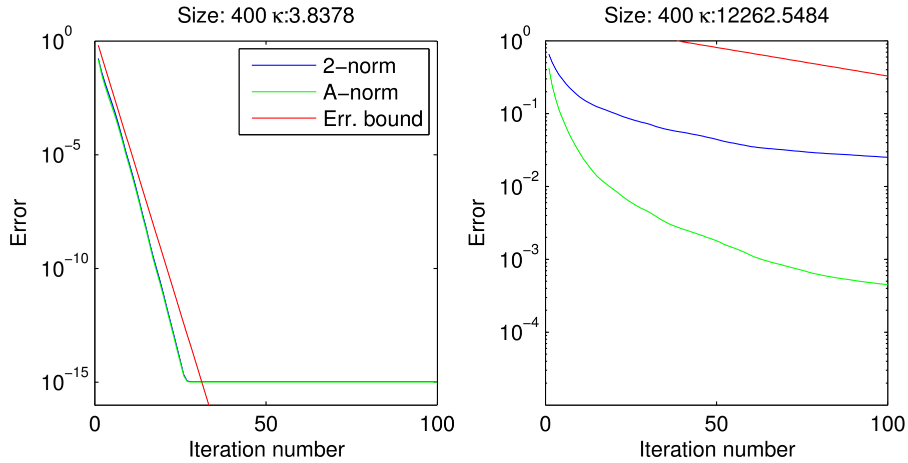

Figure 1: Convergence in 2-norm, A-norm and the error bound stated in the problem. Size of system = 400.

```
    beta = (g'*Ap)./ (p'*Ap);
    p = -g + beta*p;
    err2(loop) = sqrt((x-xsol)'*(x-xsol))/Norm2;
    errA(loop) = sqrt((x-xsol)'*A*(x-xsol))/NormA;
end
semilogy(1:ndim, err2,1:ndim,errA,'r');
legend('2-norm', 'A-norm');
xlabel('Iteration␣number'); ylabel('Error');
Tittel = ['npot=␣' num2str(npot) '␣\kappa=',num2str(kappa)];
title(Tittel);
```

a) Implement and plot the error bound

$$\|x_k - x^*\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|x_0 - x^*\|_A.$$

in the MATLAB code above. How does this compare with the actual decrease of the error? N&W say: *"This bound often gives a large overestimate"*. Is this true?

Solution: Before the loop we introduce

```
    errBound= (sqrt(kappa)-1)/(sqrt(kappa)+1);
```

and in the loop the errorbound is computed along with the others:

```
    err2(loop) = sqrt((x-xsol)'*(x-xsol))/Norm2;
    errA(loop) = sqrt((x-xsol)'*A*(x-xsol))/NormA;
    errB(loop) = 2*(errBound^loop)*NormA;
```

One example is shown in Fig. 1. Conclusions are left to the investigator!

b) Modify the well-conditioned matrix A so that it has m large eigenvalues ($3 \leq m \leq 6$) by adding a random rank-m matrix LL^{T},

$$A = (R^{\mathrm{T}}R)^{\texttt{npot}} + \mu LL^{\mathrm{T}}, \quad \mu \gg 1,$$

where L is $n \times m$ and consists of just m random column vectors. Test the performance of the CG method in this case.

Hint: Read about this in N&W p. 115–117 and the note on the web page.

Solution:
The matrix is generated simply as

```
ndim = 100; R = randn(ndim);
npot = 0.1;
mu = 100; % much larger than 1
L = randn(ndim,5);
A = (R'*R)^npot + mu*L*L';
```

An example is shown in Fig. 2.

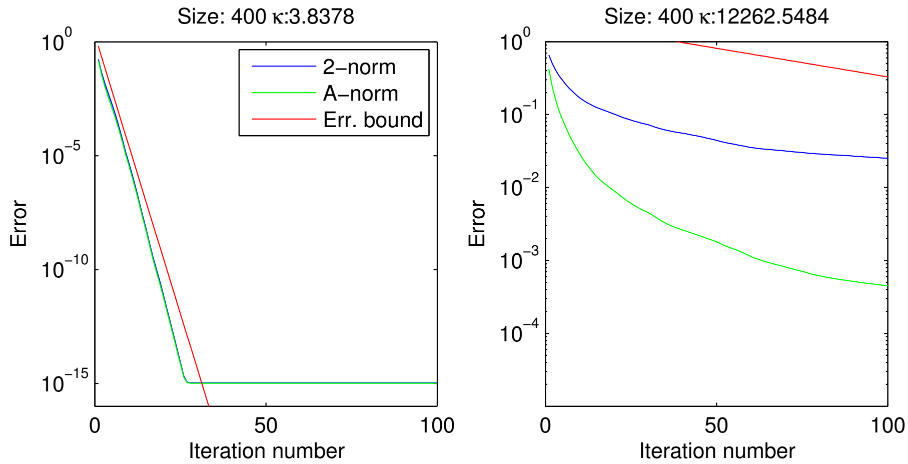

Figure 2: Convergence for a 400×400 matrix where the eigenvalues are clustered: All except 5 are clustered around 1, and the largerst 5 are about 5×10^5.

c) It is stated in the classic book by Luenberger (and also reproduced in the note) that in case **b)** above, the CG method should be restarted with a SD step every m-th step. Is this really necessary? (The SD step is obtained by setting $\beta = 0$ every m-th step).

Solution: Try yourself!