# ACM106a - Homework 4 Solutions

prepared by Svitlana Vyetrenko

November 17, 2006

1. **Problem 1:**

   (a) Let $A$ be a normal triangular matrix. Without the loss of generality assume that $A$ is upper triangular, i.e.

   $$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}$$

   Assume that $a_{1j} \neq 0$ for $j = 2 \dots n$. Then the $(1,1)$ element of $A^*A$ is equal to $a_{11}^2 + a_{12}^2 + \dots + a_{1n}^2$. The $(1,1)$ element of $AA^*$ is equal to $a_{11}^2$. Since $A$ is normal, $a_{11}^2 + a_{12}^2 + \dots + a_{1n}^2 = a_{11}^2$ and, therefore, $a_{1j} = 0$ for $j = 2 \dots n$.

   Now assume that $a_{2j} \neq 0$ for $j = 3 \dots n$. Using the above proven fact that $a_{12} = 0$ we can compare the (2,2) entries of $A^*A$ and $AA^*$ to show that all $a_{2j} = 0$ for $j = 3 \dots n$.

   Similarly, proceeding row by row and comparing the diagonal entries of $A^*A$ and $AA^*$ we can see that in order for an upper triangular $A$ to be normal, it has to be diagonal.

   (b) Any $n \times n$ matrix $A$ can be represented as $A = UTU^*$, where $T$ is upper triangular and $U$ is unitary.

   Assume that $A$ is normal, then $A^*A = AA^*$ and we have $UT^*TU^* = UTT^*U^*$, thus, $T^*T = TT^*$. By part (a) we know that if an upper triangular matrix $T$ is normal, then it is diagonal. Since the diagonal entries of the Schur form $T$ are the eigenvalues of $A$ and $T$ is a diagonal matrix, then $A = UTU^*$ gives an eigenvalue decomposition of $A$. Thus, $A$ has $n$ orthogonal eigenvectors.

   Now suppose $A$ has $n$ orthogonal eigenvectors (denote the matrix of these eigenvectors by $U$), then it can be represented as $A = U^*DU$, where $D$ is a diagonal matrix. Then by employing the fact that any diagonal matrix has to be normal we have $A^*A = U^*D^*UU^*DU = U^*D^*DU = U^*DD^*U = U^*DUU^*D^*U = AA^*$. Thus, $A$ is normal.

2. **Chapter 25, problem 25.3:**

   (a) (a) can be obtained by a sequence of left multiplications, but not by a sequence of left- and right-multiplication by matrices $Q_j$ (examples on pp.196-197 illustrate this).

   (b) (b) can be obtained by both a sequence of left multiplications and left- and right-multiplication by matrices $Q_j$ (again, see pp.196-197 for an example).

   (c) (c) can be obtained by a sequence of left multiplications by $Q_j$ (for example, consider a 3*3 matrix of rank 2).

3. **Chapter 27, problem 27.5:**

   Since $A$ is symmetric, it has a basis of orthonormal eigenvectors $q_1, q_2 \dots q_m$. Let $\lambda_1, \lambda_2 \dots \lambda_m$ be the corresponding eigenvalues. Also assume that $\lambda_1$ is the eigenvalue that is much smaller than the others in absolute value. The goal of the problem is to investigate what happens if $\lambda_1$ is very close to the shift $\mu$ and $A - \mu I$ is very ill-conditioned.

Let $\mu = \lambda_1 + \epsilon$, where $\epsilon$ is very small. Then using the computations on p.95 after $k$ inverse iterations we have:

$$(A - \mu I + \delta A)\tilde{w}_{k+1} = v_k$$

$$(A - \lambda_1 I - \epsilon I + \delta A)\tilde{w}_{k+1} = v_k \tag{1}$$

Divide both the left and the right hand sides by $\|\tilde{w}_{k+1}\|$:

$$(A - \lambda_1 I - \epsilon I + \delta A)\frac{\tilde{w}_{k+1}}{\|\tilde{w}_{k+1}\|} = \frac{v_k}{\|\tilde{w}_{k+1}\|}$$

Let $v_{k+1} = \frac{\tilde{w}_{k+1}}{\|\tilde{w}_{k+1}\|}$. Then

$$(A - \lambda_1 I)v_{k+1} = (\epsilon I - \delta A)v_{k+1} + \frac{v_k}{\|\tilde{w}_{k+1}\|}$$

Because $\|v_{k+1}\| = \|v_k\| = 1$, we have:

$$\|(A - \lambda_1 I)v_{k+1}\| \leq (|\epsilon| + \|\delta A\|) + \frac{1}{\|\tilde{w}_{k+1}\|}$$

Therefore, if $\|\tilde{w}_{k+1}\|$ is large, then $v_{k+1} = \frac{\tilde{w}_{k+1}}{\|\tilde{w}_{k+1}\|}$ gives a good approximation of the eigenvector $q_1$.

Because $q_i$ form a basis of $\mathbb{R}^m$, for some $\alpha_i$ and $\beta_i$ we have the following representations:

$$v_k = \sum \alpha_i q_i$$

$$\tilde{w}_{k+1} = \sum \beta_i q_i$$

After plugging this into (1) we have:

$$(A - \lambda_1 I)\sum \beta_i q_i - \epsilon \sum \beta_i q_i + \delta A \sum \beta_i q_i = \sum \alpha_i q_i \tag{2}$$

Because $(A - \lambda_1 I)\sum \beta_i q_i = \sum (\lambda_i - \lambda_1)\beta_i q_i$ after multiplication of (2) by $q_1^T$ on the left we have:

$$-\epsilon\beta_1 + q_1^T \delta A \tilde{w}_{k+1} = \alpha_1$$

Therefore,

$$\begin{aligned} |\alpha_1| &\leq |\epsilon||\beta_1| + \|\delta A\|\|\tilde{w}_{k+1}\| \\ &\leq |\epsilon|\|\tilde{w}_{k+1}\| + \|\delta A\|\|\tilde{w}_{k+1}\| \end{aligned}$$

Thus,

$$\|\tilde{w}_{k+1}\| \geq \frac{|\alpha_1|}{|\epsilon| + \|\delta A\|} \tag{3}$$

Hence, $\|\tilde{w}_{k+1}\|$ is large and despite ill-conditioning inverse iteration produces an accurate solution.

**Note:** For this problem I used Wilkinson, The algebraic eigenvalue problem, 1965.

4. **Problem 4:**

(a)
```
function [lambda,v]=rayleigh_quotient(A, num_iter,starting_vector)
 n=size(A);
 v=starting_vector;
 v=v/norm(v);
 lambda=v'*A*v;
 for i=1:num_iter
     w=(A-lambda*eye(n))\v;
     v=w/norm(w);
     lambda=v'*A*v;
 end;
```

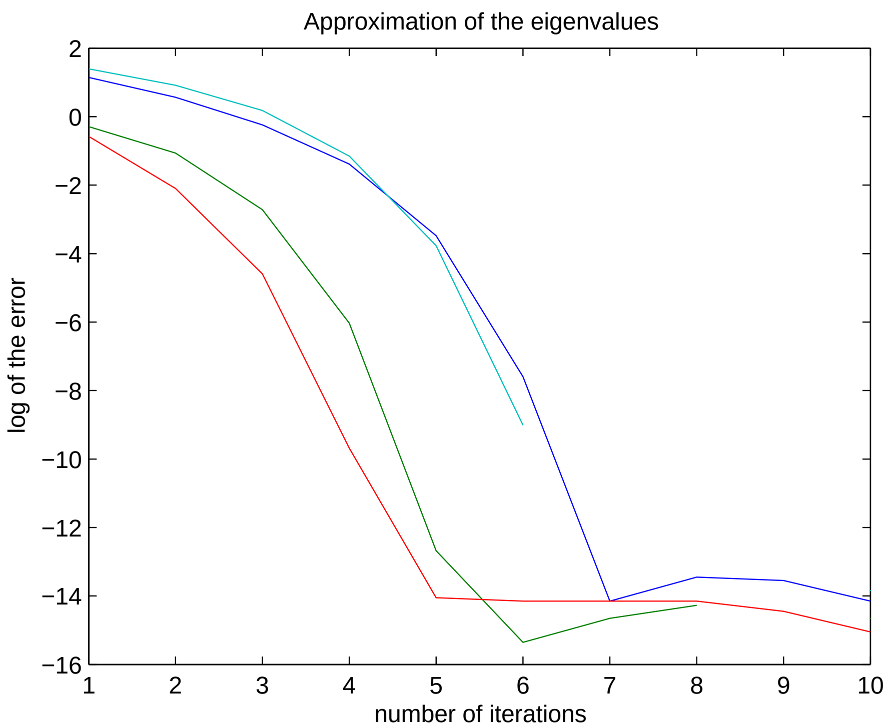
(b) Let $S = U\Sigma V^*$ be the singular value decomposition of $S$. We know that $Cond(S) = \frac{\sigma_m ax(s)}{\sigma_m in(S)}$. Therefore, if we fix the ratio of maximum to minimum singular value of $S$ to be equal to 20, and generate the intermediate singular values and orthogonal matrices $U$ and $V$ randomly (for example, by running the QR factorization on randomly generated $4 \times 4$ matrices to ensure orthogonality), then $Cond(S) = 20$.

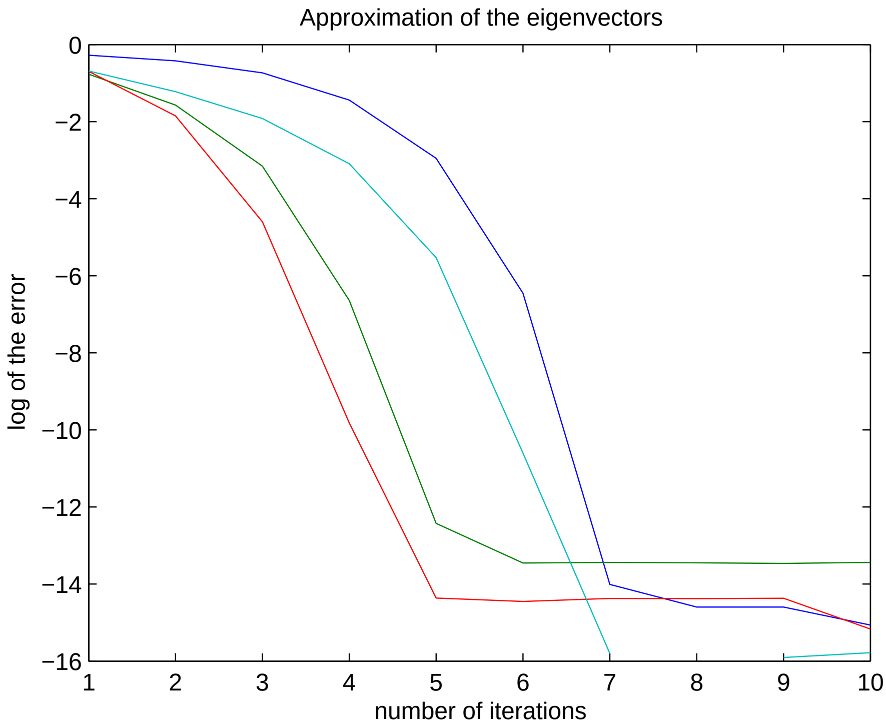
Example of the code:

```
function S=gen_s;
 %generate entries in the range 2..19
 rand1=rand(1)*17+2;
 rand2=rand(1)*17+2;

 Sigma=diag([20 rand1 rand2 1]);
 U_rand=rand(4,4);
 V_rand=rand(4,4);

 [U_orth, R1]=qr(U_rand);
 [V_orth, R2]=qr(V_rand);
 S=U_orth*Sigma*V_orth;
```

(c)
```
function speed_of_conv
%obtained by S=gen_s;
S= [-1.62886147322250   4.24469694941438  -4.08799056407617  -4.90119487824613
    -6.78153604613332   8.55038958204748   3.55067404752499  -5.55946396709622
     5.19316051425327  15.54586417118852  -8.43124975651081   0.25890869393220
    -7.76317650562957   1.71798066224471   0.30131439216199  -6.48956716145272];
A=S*diag([1 2 6 30])*inv(S);
F=S;

%for lambda=1
starting_vector=[0.32708034490656; 0.47620926031277; 0.22598334525969; 0.96200619568934];
for num_iter=1:10
    [lambda,v]=rayleigh_quotient(A,num_iter,starting_vector);
    error_1_lambda(num_iter)=abs(1-lambda);
    if (F(1,1)*v(1)<0) v=-v;
    end;
    error_1_v(num_iter)=norm(v-F(:,1)/norm(F(:,1)));
end;

%for lambda=2
starting_vector=[0.30645822520195; 0.77892558525256; 0.89534244333904; 0.25372331901581];
for num_iter=1:10
    [lambda,v]=rayleigh_quotient(A,num_iter,starting_vector);
```

```
    error_2_lambda(num_iter)=abs(2-lambda);
    if (F(1,2)*v(1)<0) v=-v;
    end;
    error_2_v(num_iter)=norm(v-F(:,2)/norm(F(:,2)));
end;

%for lambda=6
 starting_vector=[0.54423875980860; 0.55844540041460; 0.66968372236257; 0.58392692252473]
 for num_iter=1:10
    [lambda,v]=rayleigh_quotient(A,num_iter,starting_vector);
    error_3_lambda(num_iter)=abs(6-lambda);
    if (F(1,3)*v(1)<0) v=-v;
    end;
    error_3_v(num_iter)=norm(v-F(:,3)/norm(F(:,3)));
end;

%for lambda=30
starting_vector=[0.43192792375934; 0.43609798620903; 0.22430472608039; 0.01318434025798];
for num_iter=1:10
    [lambda,v]=rayleigh_quotient(A,num_iter,starting_vector);
    error_4_lambda(num_iter)=abs(30-lambda);
    if (F(1,4)*v(1)<0) v=-v;
    end;
    error_4_v(num_iter)=norm(v-F(:,4)/norm(F(:,4)));
end;

plot(1:10,log10(error_1_lambda),1:10,log10(error_2_lambda),1:10,log10(error_3_lambda),1:10
plot(1:10,log10(error_1_v),1:10,log10(error_2_v),1:10,log10(error_3_v),1:10,log10(error_4_
```






(d) We can deduce from the graphs that for a non-symmetric case the speed of convergence of both eigenvalues and eigenvectors is roughly quadratic.