Assignment 6: Rayleigh Quotient Iteration; Splitting Methods

Gregory Smetana
ID 1917370
ACM 106a

November 21, 2013

1 The spectral radius

The spectral radius of the matrix A is defined as

$$\rho(A) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\} \tag{1}$$

a)

$$\begin{aligned} \|A\|_2 &= \max_{\|x\|\neq 0} \frac{\|Ax\|_2}{\|x\|_2} \\ &= \max_{\|x\|\neq 0} \frac{\sqrt{x^* A^* A x}}{\|x\|_2} \end{aligned} \tag{2}$$

The matrix A^*A is Hermitian, so there exists an eigendecomposition $A^*A = QDQ^*$ with $D = \text{diag}(\lambda_1, ..., \lambda_n)$, where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$.

$$\begin{aligned} \|A\|_2 &= \max_{\|x\|\neq 0} \frac{\sqrt{x^* QDQ^* x}}{\|x\|_2} \\ &= \max_{\|x\|\neq 0} \frac{\sqrt{x^* QDQ^* x}}{\|x\|_2} \\ &= \max_{\|x\|\neq 0} \frac{\sqrt{(Q^*x)^* DQ^* x}}{\|Qx\|_2} \\ &= \max_{\|y\|\neq 0} \frac{\sqrt{y^* D y}}{\|y\|_2} \\ &= \max_{\|y\|\neq 0} \sqrt{\frac{\sum \lambda_i y_i^2}{\sum y_i^2}} \\ &= \sqrt{\lambda_1(A^*A)} = \sigma_1(A) \end{aligned} \tag{3}$$

Thus we have shown that $\|A\|_2 = \sigma_1(A)$. From the last line above, it was shown that

$$\|A\|_2 = \sqrt{\rho(A^*A)} \tag{4}$$

For the special case of A symmetric, the eigenvalues of $A^*A = A^2 = QD^2Q^*$ are the squares of the eigenvalues of A, so

$$\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{\rho(A)^2} = \rho(A) \tag{5}$$

Therefore,

$$\boxed{\rho(A) = \sigma_1(A) = \|A\|_2} \tag{6}$$

b)

The statement $\|A\|_2 = \rho(A)$ is false for a general matrix A. In particular, we examine the matrix

$$A = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix} \tag{7}$$

$$0 = \det \begin{pmatrix} 1-\lambda & m \\ 0 & 1-\lambda \end{pmatrix} = (1-\lambda)^2 \tag{8}$$

so we have $\rho(A) = 1$

$$\|A\|_2 = \max_{\|x=1\|} \|Ax\|_2 \tag{9}$$

$$\|A\|_2 \geq \| \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \|_2 = \| \begin{pmatrix} m \\ 1 \end{pmatrix} \|_2 = \sqrt{1+m^2} \tag{10}$$

Therefore A satisfies $\|A\|_2 > m$ and $\rho(A) < \|A\|_2$ if $m \geq 1$. We have shown that we may have $\rho(A) < \|A\|_2$ if A is not Hermitian.

c)

$$Ax = \lambda x \tag{11}$$

Multiplying by A on both sides and substituting $A = \lambda x$ on the right, we have

$$A^r x = \lambda^r x \tag{12}$$

$$\|A^r x\| = |\lambda^r| \|x\| \tag{13}$$

for an operator norm, $\|A^r x\| \leq \|A^r\| \|x\|$, so

$$\|A^r\| \|x\| \geq |\lambda^r| \|x\| \tag{14}$$

$$\|A^r\| \geq |\lambda^r| \tag{15}$$

therefore

$$\boxed{\rho(A) \leq \|A^r\|^{1/r}} \tag{16}$$

for any operator norm $\|\cdot\|$ and positive r

d)

A normal matrix is diagonalizable. Therefore, we may express

$$A = QDQ^* = \sum_{i}^{n} \lambda_i q_i q_i^* \tag{17}$$

with $D = \text{diag}(\lambda_1, ..., \lambda_n)$, $\lambda_i \in \mathbf{R}$ and eigenvectors q_i normalized such that $\|q_i\| = 1$

$$A^r = \sum_{i=1}^{n} \lambda_i^r q_i q_i^* \tag{18}$$

Now consider the norm of A^r divided by the spectral radius with $r \to \infty$

$$\lim_{r\to\infty} \frac{\|A^r\|}{\rho(A)^r} = \lim_{r\to\infty} \| \sum_{i=1}^{n} \frac{\lambda_i^r q_i q_i^*}{\rho(A)^r} \| \tag{19}$$

The terms may be grouped by unique magnitude of eigenvalue. Say that there are $m \leq n$ unique magnitudes $|\lambda_i|$ with k_i associated terms in the sum. Using norm inequalities and Equation 1 to write $\rho(A) = |\lambda_{max}|$, we may rewrite the sum:

$$\lim_{r\to\infty} \frac{\|A^r\|}{\rho(A)^r} \leq \lim_{r\to\infty} \sum_{i=1}^{m} \frac{|\lambda_i^r|}{|\lambda_{max}^r|} \| \sum_{j=1}^{k_i} q_{i,j} q_{i,j}^* \| \tag{20}$$

Now we will show that

$$\| \sum_{j=1}^{k} q_j q_j^* \| = 1 \tag{21}$$

Starting with the definition of the norm

$$\| \sum_{j=1}^{k} q_j q_j^* \| = \max_{\|x\|=1} \| \sum_{j=1}^{k} q_j q_j^* x \| \tag{22}$$

Expressing x in the basis of n eigenvectors, $x = \sum_{i=1}^{n} \alpha_i q_i$

$$\| \sum_{j=1}^{k} q_j q_j^* \| = \max_{\|x\|=1} \| \sum_{j=1}^{k} \alpha_j q_j \| \tag{23}$$

$$\| \sum_{j=1}^{k} \alpha_j q_j \| \leq \| \sum_{i=1}^{n} \alpha_i q_i \| = \|x\| = 1 \tag{24}$$

so it is maximized when x is written in the basis of k eigenvectors, $x = \sum_{j=1}^{k} \alpha_j q_j$. We see that

$$\| \sum_{j=1}^{k} q_j q_j^* \| = 1 \tag{25}$$

Therefore,

$$\lim_{r\to\infty} \frac{\|A^r\|}{\rho(A)^r} \leq \lim_{r\to\infty} \sum_{i=1}^{m} \frac{|\lambda_i^r|}{|\lambda_{max}^r|} \tag{26}$$

As $r \to \infty$, only the term corresponding to the maximum eigenvalue magnitude will not vanish, so

$$\lim_{r\to\infty} \frac{\|A^r\|}{\rho(A)^r} \leq 1 \tag{27}$$

Combined with the previous result which stated

$$\frac{\|A^r\|}{\rho(A)^r} \geq 1 \tag{28}$$

we have shown that

$$\frac{A^r}{\rho(A)^r} \to 1 \tag{29}$$

Therefore

$$\boxed{\rho(A) = \lim_{r\to\infty} \|A^r\|^{1/r}} \tag{30}$$

2 Convergence of the Jacobi and Gauss-Seidel methods

A splitting method converges if and only if the update matrix R has spectral radius satisfying

$$\rho(R) < 1 \tag{31}$$

The update matrix of the Jacobi method is

$$R_J = L + U \tag{32}$$

The update matrix of Gauss-Seidel is

$$R_{GS} = (I - L)^{-1} U \tag{33}$$

where we adopt the Demmel notation of

$$A = D - \tilde{L} - \tilde{U} = D(I - L - U) \tag{34}$$

where $-\tilde{L} = -DL$ is the strictly lower triangular part of A and $-\tilde{U} = -UL$ is the strictly upper triangular part of A.

a)

Suppose $Ax = b$, where $A \in \mathbf{R}^{2\times 2}$ is symmetric positive definite. This means A is symmetric and $x^T Ax > 0$ for all $x \in \mathbf{R}^2$, or equivalently, all eigenvalues of A are positive.

Consider

$$A = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{35}$$

The eigenvalues of A are found with the characteristic polynomial:

$$0 = (a-\lambda)(c-\lambda) - b^2 \tag{36}$$

$$\lambda = \frac{1}{2}\left(a + c \pm \sqrt{a^2 + 4b^2 - 2ac + c^2}\right) \tag{37}$$

The eigenvalues must be positive, so we see the trace and determinant must be positive:

$$a + c > 0 \tag{38}$$

$$ac - b^2 > 0 \tag{39}$$

For this matrix, we have

$$L = \begin{pmatrix} 0 & 0 \\ -b/c & 0 \end{pmatrix} \tag{40}$$

$$U = \begin{pmatrix} 0 & -b/a \\ 0 & 0 \end{pmatrix} \tag{41}$$

$$R_J = L + U = \begin{pmatrix} 0 & -b/a \\ -b/c & 0 \end{pmatrix} \tag{42}$$

Solving for the eigenvalues of R_J,

$$\lambda_J^2 - \frac{b^2}{ac} = 0 \tag{43}$$

$$\lambda_J = \pm\frac{b}{\sqrt{ac}} \tag{44}$$

The spectral radius shows that the Jacobi method converges for the matrix A:

$$\boxed{\rho(R_J) = b^2/ac < 1} \tag{45}$$

Using the identity of the inverse of a 2×2 matrix to find R_{GS},

$$\begin{aligned} R_{GS} &= (I-L)^{-1}U \\ &= \begin{pmatrix} 1 & 0 \\ b/c & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 & -b/a \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ -b/c & 1 \end{pmatrix} \begin{pmatrix} 0 & -b/a \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -b/a \\ 0 & b^2/ac \end{pmatrix} \end{aligned} \tag{46}$$

Solving for the eigenvalues of R_{GS},

$$\lambda_{GS} = \{0, b^2/ac\} \tag{47}$$

The spectral radius shows that the Gauss-Seidel method converges:

$$\boxed{\rho(R_{GS}) = b^2/ac < 1} \tag{48}$$

b)

Now consider the 2×2 matrix

$$A = \begin{pmatrix} 1 & \alpha \\ -\alpha & 1 \end{pmatrix} \tag{49}$$

For this matrix,

$$L = \begin{pmatrix} 0 & 0 \\ \alpha & 0 \end{pmatrix} \tag{50}$$

$$U = \begin{pmatrix} 0 & -\alpha \\ 0 & 0 \end{pmatrix} \tag{51}$$

$$R_J = L + U = \begin{pmatrix} 0 & -\alpha \\ \alpha & 0 \end{pmatrix} \tag{52}$$

Computing the eigenvalues of R_J,

$$\lambda_J = \pm i\alpha \tag{53}$$

The spectral radius shows that the Jacobi method converges when

$$\boxed{\rho(R_J) = |\alpha| < 1} \tag{54}$$

$$\begin{aligned} R_{GS} &= (I - L)^{-1} U \\ &= \begin{pmatrix} 1 & 0 \\ -\alpha & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 & -\alpha \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ \alpha & 1 \end{pmatrix} \begin{pmatrix} 0 & -\alpha \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -\alpha \\ 0 & -\alpha^2 \end{pmatrix} \end{aligned} \tag{55}$$

Computing the eigenvalues of R_{GS},

$$\lambda_{GS} = \{0, -\alpha^2\} \tag{56}$$

The spectral radius shows that the Gauss-Seidel method converges when

$$\boxed{\rho(R_{GS}) = |\alpha^2| < 1} \tag{57}$$

c)

Now consider the 2×2 block matrix

$$A = \begin{pmatrix} I & S \\ -S^T & I \end{pmatrix} \tag{58}$$

For this matrix,

$$L = \begin{pmatrix} 0 & 0 \\ S^T & 0 \end{pmatrix} \tag{59}$$

$$U = \begin{pmatrix} 0 & -S \\ 0 & 0 \end{pmatrix} \tag{60}$$

$$R_J = L + U = \begin{pmatrix} 0 & -S \\ S^T & 0 \end{pmatrix} \tag{61}$$

Computing the eigenvalues of R_J,

$$\det \begin{pmatrix} -\lambda_J I & -S \\ S^T & -\lambda_J I \end{pmatrix} = 0 \tag{62}$$

We may use an identity for the determinant of a block matrix:

$$\det \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det(AD - BC) \tag{63}$$

where the blocks are square matrices and C and D commute ($CD = DC$)

$$0 = \det(\lambda_J^2 I + SS^T) \tag{64}$$

Using the singular value decomposition of S,

$$S = U \Sigma V^T \tag{65}$$

where

- $U \in \mathbf{R}^{n \times n}$ such that $U^T U = I$
- $V \in \mathbf{R}^{n \times n}$ such that $V^T V = I$
- $\Sigma = \text{Diag}(\sigma_1, \sigma_2, ..., \sigma_n)$ where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n \geq 0$.

we have

$$\begin{aligned} 0 &= \det(\lambda_J^2 I + U \Sigma^2 U^T) \\ &= \det(U(\lambda_J^2 I + \Sigma^2) U^T) \\ &= \det(U) \det(\lambda_J^2 I + \Sigma^2) \det(U^T) \end{aligned} \tag{66}$$

The determinant of a unitary matrix is ± 1, so we must have

$$0 = \det(\lambda_J^2 I + \Sigma^2) = \det \begin{pmatrix} \lambda_J^2 + \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \lambda_J^2 + \sigma_n^2 \end{pmatrix} \tag{67}$$

with

$$\lambda_{J,i} = \pm i \sigma_i \tag{68}$$

The spectral radius shows that the Jacobi method converges when

$$\boxed{\rho(R_J) = \sigma_1 < 1} \tag{69}$$

Now examining the Gauss-Seidel algorithm,

$$\begin{aligned} R_{GS} &= (I - L)^{-1} U \\ &= \begin{pmatrix} I & 0 \\ -S^T & I \end{pmatrix}^{-1} \begin{pmatrix} 0 & -S \\ 0 & 0 \end{pmatrix} \end{aligned} \tag{70}$$

Using the blockwise inversion formula

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{pmatrix} \tag{71}$$

with $B = 0$, $C = -S^T$, and $A = D = I$, we have

$$\begin{aligned} R_{GS} &= \begin{pmatrix} I & 0 \\ S^T & I \end{pmatrix} \begin{pmatrix} 0 & -S \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -S \\ 0 & -S^T S \end{pmatrix} \end{aligned} \tag{72}$$

Computing the eigenvalues of R_{GS},

$$\begin{aligned} 0 &= \det \begin{pmatrix} -\lambda_{GS} I & -S \\ 0 & -S^T S - \lambda_{GS} I \end{pmatrix} \\ &= \det \left(\lambda_{GS} S^T S + \lambda_{GS}^2 I \right) \end{aligned} \tag{73}$$

Using the singular value decomposition $S = U \Sigma V^T$,

$$\begin{aligned} 0 &= \det \left(\lambda_{GS} V \Sigma^2 V^T + \lambda_{GS}^2 I \right) \\ &= \det \left(V \left(\lambda_{GS} \Sigma^2 + \lambda_{GS}^2 I \right) V^T \right) \\ &= \det(V) \det \left(\lambda_{GS} \Sigma^2 + \lambda_{GS}^2 I \right) \det(V^T) \end{aligned} \tag{74}$$

Since the determinant of a unitary matrix is ± 1, we must have

$$0 = \det\left(\lambda_{GS}\Sigma^2 + \lambda_{GS}^2 I\right) = \det\begin{pmatrix} \lambda_{GS}(\lambda_{GS}+\sigma_1^2) & & 0 \\ & \ddots & \\ 0 & & \lambda_{GS}(\lambda_{GS}+\sigma_n^2) \end{pmatrix} \tag{75}$$

so we have

$$\lambda_{GS} = \{0, -\sigma_i^2\} \tag{76}$$

The spectral radius shows that the Gauss-Seidel method converges when

$$\boxed{\rho(R_{GS}) = \sigma_1^2 < 1} \tag{77}$$

3 Rayleigh Quotient Iteration

a)

Rayleigh Quotient iteration as described in Demmel Algorithm 5.1 was implemented in Matlab in `rayleigh_quotient.m`. The “\” solver was used to solve linear systems.

b)

A random 4×4 orthogonal matrix U was used to generate

$$A = UDU^T \tag{78}$$

where $D = \text{Diag}(2, 4, 13, 27)$. The starting point for each Rayleigh Quotient iteration was chosen as 10 times the one column of U plus a random combination of the other columns. This led to convergence of each eigenvector of A, which was verified by examination of the Rayleigh Quotients.

The error was measured as the forward error between the current Rayleigh quotient and desired eigenvalue. The error on a log scale as a function of the number of iterations is displayed in Figure 1.

The error at each step of the algorithm behaves like

$$e = c^{-r^k} \tag{79}$$

where c is some constant and r is the rate of convergence. so we have

$$\log(\log(e)) = \log(\log(c^{-r^k})) = \log(-r^k \log(c)) = k\log(-r) + \log(\log(c)) \tag{80}$$

From the log-log plot in Figure 2, we see that $\log(r) \sim 0.4$. Therefore the Rayleigh Quotient iteration converges roughly at a rate $r = 2.5$.

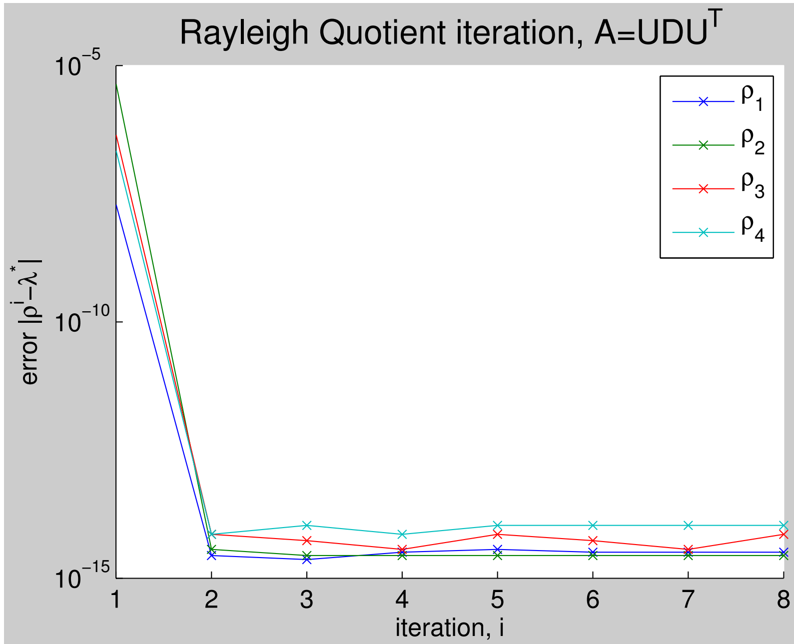

Figure 1

c)

A random 4×4 matrix S with $\kappa(S) = \sigma_1(S)/\sigma_4(S) = 10$ was generated by taking the product of its singular value decomposition. Then Rayleigh Quotient iteration was performed on

$$A = SDS^{-1} \tag{81}$$

The error on a log scale as a function of the number of iterations is displayed in Figure 3. From the log-log plot in Figure 4, we see that $\log(r) \sim 0.25$, and the Rayleigh Quotient iteration converges roughly at a rate $r = 1.8$.

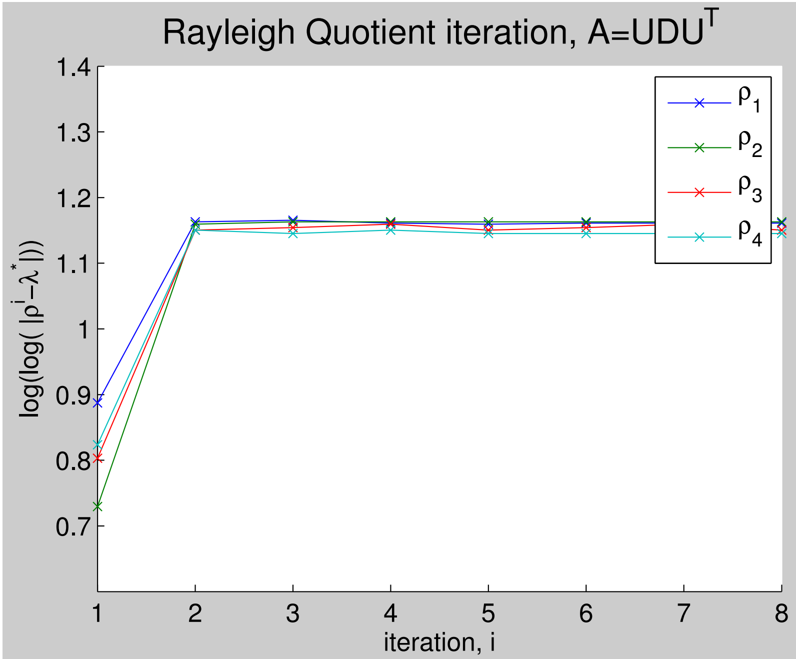

Figure 2

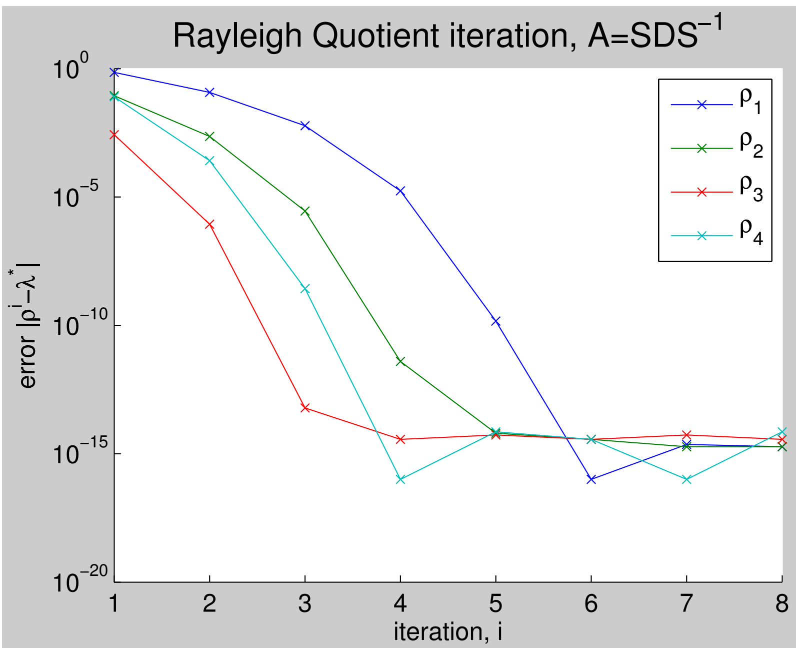

Figure 3

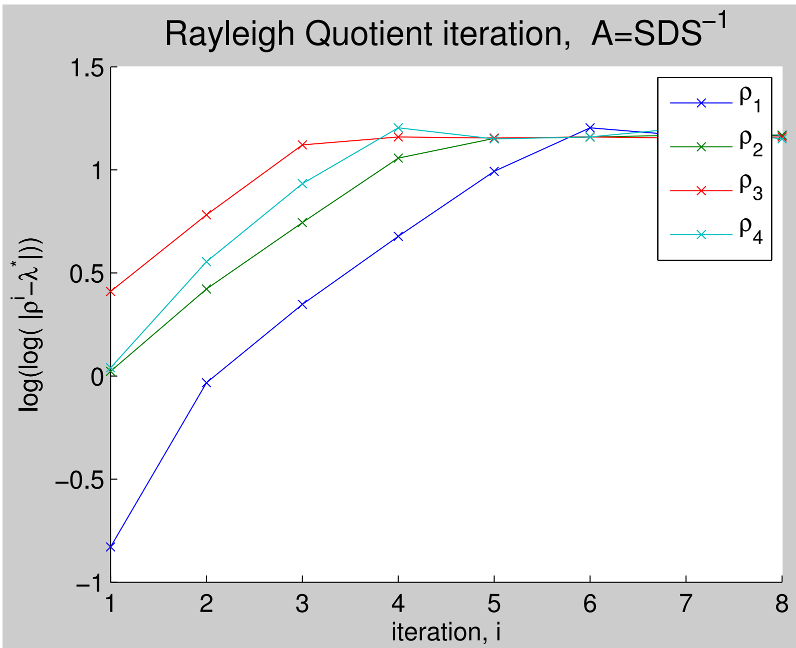
Rayleigh Quotient iteration, A=SDS^{-1}
log(log(|ρ^{i}−λ*|))
iteration, i
ρ$_1$
ρ$_2$
ρ$_3$
ρ$_4$
1.5
1
0.5
0
−0.5
−1
1
2
3
4
5
6
7
8

Figure 4

A rayleigh_quotient.m

```
function [ rho, x , error] = rayleigh_quotient( A, x0, D)
%UNTITLED3 Summary of this function goes here
%   Detailed explanation goes here

rho = x0'* A *x0/(x0'*x0);
i = 1;
x = x0;
n = size(A,1);
error=zeros(8,1);
while i <= 8
    y = (A - rho*eye(n))\x;
    x = y / norm(y);
    rho = x'*A *x / (x'*x);
    error(i) = min(abs(rho-D)) + 10^-16;
    i = i+1;
end
```

B Smetana_Gregory_1917370_A6_P3_DIARY.txt

```
run('Smetana_Gregory_1917370_A6_P3.m');
(warnings removed)
part a

ans =

   2.0000e+00   4.0000e+00   1.3000e+01   2.7000e+01

(warnings removed)
ans =

   2.0000e+00   4.0000e+00   1.3000e+01   2.7000e+01

diary off
```

C Smetana_Gregory_1917370_A6_P3m

```
%Smetana_Gregory_1917370_A6_P3
clear;
clc;
close all;
path(path,'export_fig/');

%% part a
```

```
U = orth(rand(4,4));
D = diag([2,4,13,27]);
A = U * D * U';

x1 = 10*U(:,1)+ rand*U(:,2)+rand*U(:,3)+rand*U(:,4);
x2 = rand*U(:,1)+ 10*U(:,2)+rand*U(:,3)+rand*U(:,4);
x3 = rand*U(:,1)+ rand*U(:,2)+10*U(:,3)+rand*U(:,4);
x4 = rand*U(:,1)+ rand*U(:,2)+rand*U(:,3)+10*U(:,4);

[rho1, v1, e1] = rayleigh_quotient(A, x1, diag(D));
[rho2, v2, e2] = rayleigh_quotient(A, x2, diag(D));
[rho3, v3, e3] = rayleigh_quotient(A, x3, diag(D));
[rho4, v4, e4] = rayleigh_quotient(A, x4, diag(D));

display('part a');
[rho1,rho2,rho3,rho4]

figure;
hold all;
plot(e1,'x-');
plot(e2,'x-');
plot(e3,'x-');
plot(e4,'x-');

xlabel('iteration, i','FontSize',12);
ylabel('error |\rho^i-\lambda^*|','FontSize',12);
legend('\rho_1', '\rho_2', '\rho_3', '\rho_4');
title(['Rayleigh Quotient iteration, A=UDU^T'],'FontSize',16);

set(gca,'FontSize',12);
set(gca,'yscale','log')
filename = ['report/p3b.pdf'];
export_fig(filename)

figure;
hold all;
plot(log10(-log10(e1)),'x-');
plot(log10(-log10(e2)),'x-');
plot(log10(-log10(e3)),'x-');
plot(log10(-log10(e4)),'x-');

xlabel('iteration, i','FontSize',12);
ylabel('log(log( |\rho^i-\lambda^*|))','FontSize',12);
legend('\rho_1', '\rho_2', '\rho_3', '\rho_4');
title(['Rayleigh Quotient iteration, A=UDU^T'],'FontSize',16);

set(gca,'FontSize',12);
filename = ['report/p3blog.pdf'];
export_fig(filename)
%% part b
```

```
U = orth(rand(4,4));
V = orth(rand(4,4));
sigma = diag([10, 5+5*rand, 1+4*rand, 1]);
S = U * sigma * V'; % generate matrix with condition number = 10
A = S * D * inv(S);

x1 = 10*S(:,1)+ rand*S(:,2)+rand*S(:,3)+rand*S(:,4);
x2 = rand*S(:,1)+ 10*S(:,2)+rand*S(:,3)+rand*S(:,4);
x3 = rand*S(:,1)+ rand*S(:,2)+10*S(:,3)+rand*S(:,4);
x4 = rand*S(:,1)+ rand*S(:,2)+rand*S(:,3)+10*S(:,4);

[rho1, v1, e1] = rayleigh_quotient(A, x1, diag(D));
[rho2, v2, e2] = rayleigh_quotient(A, x2, diag(D));
[rho3, v3, e3] = rayleigh_quotient(A, x3, diag(D));
[rho4, v4, e4] = rayleigh_quotient(A, x4, diag(D));

display('part b');
[rho1,rho2,rho3,rho4]

figure;
hold all;
plot(e1,'x-');
plot(e2,'x-');
plot(e3,'x-');
plot(e4,'x-');

xlabel('iteration, i','FontSize',12);
title(['Rayleigh Quotient iteration, A=SDS^-^1'],'FontSize',16);
ylabel('error |\rho^i-\lambda^*|','FontSize',12);
legend('\rho_1', '\rho_2', '\rho_3', '\rho_4');
set(gca,'FontSize',12);
set(gca,'yscale','log')
filename = ['report/p3c.pdf'];
export_fig(filename)

figure;
hold all;
plot(log10(-log10(e1)),'x-');
plot(log10(-log10(e2)),'x-');
plot(log10(-log10(e3)),'x-');
plot(log10(-log10(e4)),'x-');

xlabel('iteration, i','FontSize',12);
ylabel('log(log( |\rho^i-\lambda^*|))','FontSize',12);
legend('\rho_1', '\rho_2', '\rho_3', '\rho_4');
title(['Rayleigh Quotient iteration,  A=SDS^-^1'],'FontSize',16);

set(gca,'FontSize',12);
filename = ['report/p3clog.pdf'];
export_fig(filename)
```