

# Numerical Linear Algebra

## Programming Assignment #09

2015-17231

박우정

### Exercise 5.3.

(2.29)로 주어진 행렬과 벡터로 이루어진 방정식  $Ax = b$ 를 Householder transformation으로 푸는 matlab code는 다음과 같다. 편의상 Householder matrix의 한 column을 계산하는 함수를 정의한 후, 함수에 submatrix(subvector)를 넣는 아이디어를 활용하였다.

```
function [H]=house(a)
[n,m]=size(a);
e1=[1;zeros(1,n-1)'];
k=-sign(a(1))*norm(a);
lambda=sqrt(2*norm(a)*(norm(a)+abs(a(1))));
v=(a-k*e1)./lambda;
H=eye(n)-2*v*v';
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=input('What is the dimension n? ');
h=1/n;n=n-1;A=zeros(n);Q=eye(n);R=zeros(n);b=zeros(n,1);x=zeros(n,1);
for i=1:n
    A(i,i)=2/h^2;
    b(i,1)=0;
end
b(1,1)=1;saveb=b;
for i=1:n-1
    A(i,i+1)=-1/h^2;
    A(i+1,i)=-1/h^2;
end
saveA=A;
for i=1:n
    h=house(A(i:n,i)); %householder function.
    H=eye(n);
    H(i:n,i:n)=h;
    Q=Q*H;
    A=H*A;
end
R=A;
b=Q'*b;
for j=n:-1:1
    x(j,1)=(b(j,1)-dot(R(j,j+1:n),x(j+1:n,1)))/R(j,j);
end
for j=1:n
    if(abs(saveb(j,1)-dot(saveA(j,1:n),x(1:n,1)))>5*10^-13)
        fprintf('The computed solution seems to be wrong at %f \n',j);
    end
end
disp('The solution x is as follows: ')
disp(x)
```

$n = 100$ 을 주고 실행하면 다음과 같은 결과를 얻는다.

>> Ex5_3_Householder	0.6900	0.3400
What is the dimension n? 100	0.6800	0.3300
The solution x is as follows:	0.6700	0.3200
1.0e-04 *	0.6600	0.3100
	0.6500	0.3000
0.9900	0.6400	0.2900
0.9800	0.6300	0.2800
0.9700	0.6200	0.2700
0.9600	0.6100	0.2600
0.9500	0.6000	0.2500
0.9400	0.5900	0.2400
0.9300	0.5800	0.2300
0.9200	0.5700	0.2200
0.9100	0.5600	0.2100
0.9000	0.5500	0.2000
0.8900	0.5400	0.1900
0.8800	0.5300	0.1800
0.8700	0.5200	0.1700
0.8600	0.5100	0.1600
0.8500	0.5000	0.1500
0.8400	0.4900	0.1400
0.8300	0.4800	0.1300
0.8200	0.4700	0.1200
0.8100	0.4600	0.1100
0.8000	0.4500	0.1000
0.7900	0.4400	0.0900
0.7800	0.4300	0.0800
0.7700	0.4200	0.0700
0.7600	0.4100	0.0600
0.7500	0.4000	0.0500
0.7400	0.3900	0.0400
0.7300	0.3800	0.0300
0.7200	0.3700	0.0200
0.7100	0.3600	0.0100
0.7000	0.3500	

허용오차  $5 \times 10^{-13}$ 을 넘어가는 solution이 없고, 이전 Exercise에서 푼 바와 비교해보면 같은 solution을 얻었으므로, 알맞게 프로그래밍되었다고 할 수 있다.

#### Exercise 5.4.

(2.29)로 주어진 행렬과 벡터로 이루어진 방정식  $Ax = b$ 를 Givens Rotation으로 푸는 matlab code는 다음과 같다. G의 submatrix를  $\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ 가 되도록 부여하였다.

```
n=input('What is the dimension n? ');
h=1/n;n=n-1;A=zeros(n);Q=eye(n);R=zeros(n);b=zeros(n,1);x=zeros(n,1);
for i=1:n
    A(i,i)=2/h^2;
    b(i,1)=0;
end
b(1,1)=1;saveb=b;
for i=1:n-1
    A(i,i+1)=-1/h^2;
    A(i+1,i)=-1/h^2;
end
saveA=A;R=A;
for j=1:n
    for i=n:-1:j+1
        g=R(:,j);gnorm=sqrt(g(i-1)^2+g(i)^2);
        if gnorm>0
            c=g(i-1)/gnorm;
            s=-g(i)/gnorm;
            G=eye(n);
            G([i-1,i],[i-1,i])=[c,s;-s,c];
            R=G'*R;
            Q=Q*G;
        end
    end
end
b=Q'*b;
for j=n:-1:1
    x(j,1)=(b(j,1)-dot(R(j,j+1:n),x(j+1:n,1)))/R(j,j);
end
for j=1:n
    if(abs(saveb(j,1)-dot(saveA(j,1:n),x(1:n,1))>5*10^-13)
        fprintf('The computed solution seems to be wrong at %f \n',j);
    end
end
disp('The solution x is as follows: ')
disp(x)
```

허용오차를  $5 \times 10^{-13}$ ,  $n = 100$ 으로 동일하게 주고 위 프로그램을 실행하면 다음을 얻는다.

	0.7000	0.3500
>> Ex5_4_givens	0.6900	0.3400
What is the dimension n? 100	0.6800	0.3300
The solution x is as follows:	0.6700	0.3200
1.0e-04 *	0.6600	0.3100
	0.6500	0.3000
0.9900	0.6400	0.2900
0.9800	0.6300	0.2800
0.9700	0.6200	0.2700
0.9600	0.6100	0.2600
0.9500	0.6000	0.2500
0.9400	0.5900	0.2400
0.9300	0.5800	0.2300
0.9200	0.5700	0.2200
0.9100	0.5600	0.2100
0.9000	0.5500	0.2000
0.8900	0.5400	0.1900
0.8800	0.5300	0.1800
0.8700	0.5200	0.1700
0.8600	0.5100	0.1600
0.8500	0.5000	0.1500
0.8400	0.4900	0.1400
0.8300	0.4800	0.1300
0.8200	0.4700	0.1200
0.8100	0.4600	0.1100
0.8000	0.4500	0.1000
0.7900	0.4400	0.0900
0.7800	0.4300	0.0800
0.7700	0.4200	0.0700
0.7600	0.4100	0.0600
0.7500	0.4000	0.0500
0.7400	0.3900	0.0400
0.7300	0.3800	0.0300
0.7200	0.3700	0.0200
0.7100	0.3600	0.0100

허용오차를 넘는 solution이 없고, Exercise 5.3과 완전히 같은 결과를 얻었으므로, 알맞게 프로그래밍되었다고 할 수 있다.

Exercise 5.5.

(2.29)로 주어진 행렬과 벡터로 이루어진 행렬  $A$ 를 Gram-Schmidt orthogonalization으로 분해해주는 matlab코드는 아래와 같다.

```
n=input('What is the dimension n? ');
h=1/n;n=n-1;A=zeros(n);Q=eye(n);R=zeros(n);
for i=1:n
    A(i,i)=2/h^2;
end
for i=1:n-1
    A(i,i+1)=-1/h^2;
    A(i+1,i)=-1/h^2;
end
saveA=A;
for j=1:n
    q=A(:,j);
    for i=1:j-1
        R(i,j)=Q(:,i)'*A(:,j);
        q=q-R(i,j)*Q(:,i);
    end
    R(j,j)=norm(q);
    Q(:,j)=q/R(j,j);
end
for j=1:n
    for i=1:n
        if(abs(A(i,j)-Q(i,:)*R(:,j))>5*10^-13)
            fprintf('The computed solution seems to be wrong at %f \n',j);
        end
    end
end
disp('The Gram_Schmidt decomposition is as follows: ')
disp('Q')
disp(Q)
disp('R')
disp(R)
```

$n = 100$ 을 주어 실행하면 decomposed result Q와 R은 다음과 같이 얻을 수 있다.



그림 Result of R

허용오차를 위와 동일하게 주고, 검산해주는 programming을 무사히 통과하였으므로, 알맞게 프로그래밍되었다고 할 수 있다.