Numerical Linear Algebra Programming Assignment #09

2015-17231 박우정

Exercise 4.11.

Power Method를 이용해 주어진 행렬의 Maximum Eigenvalue와 Eigenvector를 찾아보자. 필요한 MATLAB 코드는 다음과 같다.

```
max iter=input("what is the max iter?");
A=zeros(3);
A(1,1) = -261; A(1,2) = 209; A(1,3) = -49; A(2,1) = -530; A(2,2) = 422; A(2,3) = -98; A(2,3) = -98; A(2,3) = -98; A(3,3) = -98
(3,1) = -800; A(3,2) = 631; A(3,3) = -144;
v=[1 0 0]';prev v=v;
tol=1.0e-15;
disp("Initial approximation of eigenvector is")
disp(prev v)
for iter=1:max iter
                z=A*v;
               v=z/norm(z);
               lambda=v'*A*v;
               diff_v=norm(v-prev_v);
               prev v=v;
                if (diff_v<tol)</pre>
                                disp("Converged at ")
                                 iter
                                break
                 end
end
disp("The maximum eigenvalue and eigenvector is")
lambda
if (A*v-lambda*v>tol)
                disp("Something is wrong")
```

max_iter=50으로 주고, 이를 실행하면 다음을 얻는다.

```
what is the max_iter?50
Initial approximation of eigenvector is
    1
    0
    0

Converged at

iter =
    34
The maximum eigenvalue and eigenvector is

lambda =
    10.0000

v =
    -0.2673
    -0.5345
    -0.8018
```

결과문에서 검산 구문을 잘 통과했으므로, Eigenvector와 Eigenvalue가 알맞게 구해졌다고 할 수 있다. 이 때, 위 프로그래밍 구문은 34번째 Power부터 미리 정해놓은 Tolerance=1.0e-15 이내로 값을 얻는다는 것을 확인할 수 있다.

Exercise 4.12.

이미 이전 Exercise 4.11에서 Power Method를 통해 maximum eigenvalue와 eigenvector를 구했으므로, 남은 정보들을 inverse iteration methods를 이용해 주어진 행렬의 Eigenvalue와 Eigenvector를 찾아보자. 필요한 MATLAB 구문은 아래와 같다.

```
max_iter=input("what is the max_iter?");
A=zeros(3);
 \texttt{A(1,1)} = -261; \texttt{A(1,2)} = 209; \texttt{A(1,3)} = -49; \texttt{A(2,1)} = -530; \texttt{A(2,2)} = 422; \texttt{A(2,3)} = -98; \texttt{A(3,1)} = -800; \texttt{A(3,2)} = 631; \texttt{A(3,3)} = -800; \texttt{A(3,2)} = 631; \texttt{
144;
v=[1 0 0]';prev_v=v;prev_A=A;
tol=1.0e-15;
disp("Initial approximation of eigenvector is")
%Find the maximum eigenvalue and eigenvector : Exercise 4.11
%Find the eigenvector corresponding to the smallest eigenvalue
for iter=1:max_iter
         z=T\prev_v;
          v=z/norm(z);
        diff_v=norm(v-prev_v);
        prev_v=v;
          if(diff_v<tol)</pre>
                     disp("Converged at ")
                     iter
                     break
           end
end
lambda=v'*A*v;
disp("The minimum eigenvalue and eigenvector is")
min lambda=lambda
min_v=v
if(A*min_v-min_lambda*min_v>tol)
         disp("Something is wrong")
%Find other eigenvector and eigenvalue.
v=[1 0 0]';prev_v=v;iter=0;
T=prev_A-6.5*eye(3);
\%We already know max and min of eigenvalues, which are 10 and 3, so I will use the mean(10,3)=6.5
for iter=1:max_iter
        z=T\prev_v;
        v=z/norm(z);
        diff_v=norm(v-prev_v);
        prev_v=v;
         if(diff_v<tol)</pre>
                    disp("Converged at ")
                     iter
                     break
          end
end
lambda=v'*A*v;
disp("The other eigenvalue and eigenvector is")
lambda
if(A*v-lambda*v>tol)
          disp("Something is wrong")
```

max_iter=500을 주고, 이를 실행하면 다음을 얻는다.

```
what is the max iter?500
Initial approximation of eigenvector is
    0
    0
Converged at
iter =
  104
The minimum eigenvalue and eigenvector is
min_lambda =
   3.0000
min_v =
   0.2593
   0.5185
   0.8148
The other eigenvalue and eigenvector is
lambda =
   4.0000
∨ =
   0.4614
   0.7098
   0.5323
```

따라서 minimum eigenvalue는 3이고, 남은 eigenvalue는 4이다. 검산 구문을 무사히 통과 했으므로 알맞게 프로그래밍되었다고 할 수 있으며, 특히 minimum eigenvalue의 power method는 104회 이후에 주어진 Tolerance 이내로 들어옴을 확인할 수 있다.

Exercise 4.13.

먼저 Example 4.2.1의 행렬의 maximum & minimum eigenvalue와 eigenvector를 구해보자. 필요한 MATLAB 구문을 아래와 같다.

```
n = input \mbox{("What are the dimension of the matrix n and max\_iter?")} \label{eq:normalize}
max iter=input("What are the dimension of the matrix n and max iter?")
h=1/n; iter=0; tol=1.0e-15; A=zeros(n-1);
for j=1:n-1
  A(j,j)=2/h^2;
end
for j=1:n-2
  A(j,j+1)=-1/h^2;
  A(j+1,j) = -1/h^2;
q=zeros(n-1,1); q(1)=1; %Initial approximated eigenvector q
%Using Power Method to find maximum eigenvalue and eigenvector
for iter=1:max_iter
  q=A*q;
  q=q/norm(q);
  lambda=q'*A*q;
  diff_v=norm(q-prev_q);
  prev_q=q;
  if(diff_v<tol)</pre>
      disp("Converged at iter=")
   end
end
disp("Computed maximum eigenvalue is")
max_lambda=lambda
disp("Computed maximum eigenvector is")
max_v=q
if(A*max_v-max_lambda*max_v>tol)
  disp("Something is wrong")
%Using Inverse Iterative Method to find minimum eigenvalue and eigenvector
q=zeros(n-1,1); q(1)=1; %Initial approximated eigenvector q
prev_q=q;
for iter=1:max_iter
  z=A\prev_q;
  q=z/norm(z);
  diff_v=norm(q-prev_q);
  prev_q=q;
   if(diff v<tol)</pre>
      disp("Converged at iter=")
      break
   end
end
lambda=q'*A*q;
disp("Computed maximum eigenvalue is")
min_lambda=lambda
disp("Computed maximum eigenvector is")
min v=q
if(A*min_v-min_lambda*min_v>tol)
   disp("Something is wrong")
```

먼저 n=10, max_iter=500을 주면 다음을 얻는다.

```
What are the dimension of the matrix n and max_iter?500
max_iter =
 500
Converged at iter=
iter =
 432
Computed maximum eigenvalue is
max_lambda =
390.2113
Computed maximum eigenvector is
max_v =
 0.1382
 -0.2629
  0.3618
 -0.4253
  0.4472
 -0.4253
  0.3618
 -0.2629
 0.1382
Converged at iter=
iter =
  27
Computed maximum eigenvalue is
min_lambda =
  9.7887
Computed maximum eigenvector is
min_v =
 0.1382
 0.2629
  0.3618
  0.4253
  0.4472
  0.4253
  0.3618
  0.2629
  0.1382
```

검산 구문을 통과했으므로 알맞게 구했으며, Tolerance 이내로 들어오는 iter는 각각 432, 27 power 이다.

이제 n=100일 때를 살펴보면 다음과 같다.

n =	0.0724	-0.0000	0.1170	0.0437
400	-0.0646	0.0000	0.1194	0.0395
100	0.0574	-0.0000	0.1217	0.0352
77)	-0.0508	0.0000	0.1239	0.0309
What are the	0.0447	-0.0000	0.1260	0.0265
dimension of the	-0.0391	0.0000	0.1280	0.0221
matrix n and	0.0341		0.1298	0.0177
max_iter?500	-0.0296 0.0256		0.1315 0.1331	0.0133 0.0089
	-0.0220		0.1331	0.0089
max_iter =	0.0188			0.0044
500	-0.0160	Converged at iter=	0.1358 0.1370	
300	0.0136	converged at Itel-	0.1370	
Computed maximum	-0.0115	iter =	0.1389	
eigenvalue is	0.0097	itel -	0.1389	
eigenvalue 13	-0.0081	27	0.1403	
max lambda =	0.0067	27	0.1408	
max_tambaa	-0.0056	Computed maximum	0.1411	
3.9940e+04	0.0046	eigenvalue is	0.1411	
J.JJ400104	-0.0038	erdenvarde 12	0.1414	
Computed maximum	0.0031	min lambda =	0.1414	
eigenvector is	-0.0025		0.1411	
9	0.0023	9.8688	0.1411	
max v =	-0.0017	3.0000	0.1403	
	0.0013	Computed maximum	0.1397	
0.0238	-0.0011	eigenvector is	0.1389	
-0.0473	0.0009		0.1380	
0.0703	-0.0007	min v =	0.1370	
-0.0924	0.0005	_	0.1358	
0.1134	-0.0004	0.0044	0.1345	
-0.1332	0.0003	0.0089	0.1331	
0.1514	-0.0003	0.0133	0.1315	
-0.1679	0.0002	0.0177	0.1298	
0.1826	-0.0002	0.0221	0.1280	
-0.1954	0.0001	0.0265	0.1260	
0.2061	-0.0001	0.0309	0.1239	
-0.2147	0.0001	0.0352	0.1217	
0.2213	-0.0001	0.0395	0.1194	
-0.2258	0.0000	0.0437	0.1170	
0.2284	-0.0000	0.0479	0.1144	
-0.2290	0.0000	0.0521	0.1117	
0.2278	-0.0000	0.0562	0.1090	
-0.2249	0.0000	0.0602	0.1061	
0.2205	-0.0000	0.0642	0.1031	
-0.2148	0.0000	0.0681	0.1000	
0.2078	-0.0000	0.0720	0.0968	
-0.1998	0.0000	0.0758	0.0935	
0.1909	-0.0000	0.0795	0.0901	
-0.1814	0.0000	0.0831	0.0867	
0.1714	-0.0000	0.0867	0.0831	
-0.1610	0.0000	0.0901	0.0795	
0.1504	-0.0000	0.0935	0.0758	
-0.1397	0.0000	0.0968	0.0720	
0.1292	-0.0000	0.1000	0.0681	
-0.1188	0.0000	0.1031	0.0642	
0.1087	-0.0000	0.1061	0.0602	
-0.0989	0.0000	0.1090	0.0562	
0.0896	-0.0000	0.1117	0.0521	
-0.0807	0.0000	0.1144	0.0479	

사이즈만 늘어났을 뿐, 위와 비슷한 결과를 얻었고, 알맞게 프로그래밍되었다 볼 수 있다.

다음으로 example 4.2.3을 살펴보자. 필요한 구문은 다음과 같다.

```
n=input("What are the dimension of the matrix n and max_iter?")
max_iter=input("What are the dimension of the matrix n and max_iter?")
h=1/n; iter=0; tol=1.0e-15; A=zeros(n-1);
for j=2:n-2
  A(j,j)=2/h^2;
A(1,1)=1/h^2; A(n-1,n-1)=1/h^2;
for j=1:n-2
  A(j,j+1)=-1/h^2;
  A(j+1,j)=-1/h^2;
q=zeros\left(n-1,1\right);\ q\left(1\right)=1;\ %Initial\ approximated\ eigenvector\ q
prev_q=q;
%Using Power Method to find maximum eigenvalue and eigenvector
for iter=1:max_iter
  q=A*q;
  q=q/norm(q);
  lambda=q'*A*q;
  diff_v=norm(q-prev_q);
  prev_q=q;
  if (diff_v<tol)</pre>
      disp("Converged at iter=")
      break
  end
end
disp("Computed maximum eigenvalue is")
max lambda=lambda
disp("Computed maximum eigenvector is")
max_v=q
if (A*max_v-max_lambda*max_v<tol)</pre>
  disp("Something is wrong")
end
%Using Inverse Iterative Method to find minimum eigenvalue and eigenvector
q=zeros(n-1,1); q(1)=1; %Initial approximated eigenvector q
prev q=q;
T=A-eye(n-1); %Since A is a symmetric positive-semidefinite matrix and we already know the smallest eigenvalue is 0.
Just to check it, we choose <math display="inline">\mu=1 by using Gerschgorin's circle theorem.
for iter=1:max_iter
  z=T\prev_q;
  q=z/norm(z);
  diff_v=norm(q-prev_q);
  prev_q=q;
  if(diff v<tol)</pre>
      disp("Converged at iter=")
      iter
      break
  end
end
lambda=q'*A*q;
disp("Computed minimum eigenvalue is")
min_lambda=lambda
disp("Computed minimum eigenvector is")
if(A*min_v-min_lambda*min_v<tol)</pre>
   disp("Something is wrong")
```

여기서 주어진 행렬 A는 Semi-positive symmetric matrix, singular matrix이므로 처음에 eigenvalue 0을 갖고, 다른 것들은 모두 양수임에 조심해야 한다. 따라서 Smallest eigenvalue가 0임은 당연하다. 위 구문을 0에 대응하는 eigenvector를 구하기 위한 것이다.

n=10일 때 max_iter=1000을 주면 다음을 얻는다.

```
What are the dimension of the matrix n and max_iter?10
n =
What are the dimension of the matrix n and \max_{} iter?1000
max iter =
 1000
Converged at iter=
iter =
 351
Computed maximum eigenvalue is
max_lambda =
387.9385
Computed maximum eigenvector is
max_v =
 0.0819
 -0.2357
  0.3611
  0.4714
 -0.4430
  0.3611
 -0.2357
  0.0819
Computed maximum eigenvalue is
min_lambda =
 7.8886e-31
Computed maximum eigenvector is
min_v =
  0.3333
  0.3333
  0.3333
  0.3333
  0.3333
  0.3333
  0.3333
  0.3333
  0.3333
```

0에 매우 가까운 값을 얻었고, 검산 구문을 통과하였으므로 알맞게 프로그래밍되었다.

다음은 n=100일 때의 상황을 관찰해보자.

What are the	0.1797	-0.0006	0.1005	0.1005
dimension of the	-0.1756	0.0005	0.1005	0.1005
matrix n and	0.1710	-0.0004	0.1005	0.1005
max_iter?100	-0.1660	0.0003	0.1005	0.1005
	0.1606	-0.0003	0.1005	0.1005
	-0.1550	0.0002	0.1005	0.1005
n =	0.1491	-0.0002	0.1005	0.1005
	-0.1431	0.0001	0.1005	0.1005
100	0.1369	-0.0001	0.1005	0.1005
	-0.1306	0.0001	0.1005	0.1005
What are the	0.1243	-0.0000	0.1005	
dimension of the	-0.1180	0.0000	0.1005	
matrix n and	0.1117		0.1005	
max_iter?1000	-0.1054		0.1005	
	0.0993		0.1005	
max_iter =	-0.0932		0.1005	
	0.0873		0.1005	
1000	-0.0816	Computed minimum	0.1005	
	0.0761	eigenvalue is	0.1005	
Computed maximum	-0.0707		0.1005	
eigenvalue is	0.0656	min_lambda =	0.1005	
	-0.0607		0.1005	
max lambda =	0.0561	2.0952e-27	0.1005	
_	-0.0516		0.1005	
3.9970e+04	0.0474	Computed minimum	0.1005	
	-0.0435	eigenvector is	0.1005	
Computed maximum	0.0398		0.1005	
eigenvector is	-0.0363	min v =	0.1005	
	0.0330	_	0.1005	
max v =	-0.0300	0.1005	0.1005	
=	0.0272	0.1005	0.1005	
0.0071	-0.0245	0.1005	0.1005	
-0.0213	0.0221	0.1005	0.1005	
0.0353	-0.0199	0.1005	0.1005	
-0.0491	0.0179	0.1005	0.1005	
0.0626	-0.0160	0.1005	0.1005	
-0.0758	0.0143	0.1005	0.1005	
0.0885	-0.0128	0.1005	0.1005	
-0.1007	0.0114	0.1005	0.1005	
0.1123	-0.0101	0.1005	0.1005	
-0.1233	0.0089	0.1005	0.1005	
0.1335	-0.0079	0.1005	0.1005	
-0.1431	0.0070	0.1005	0.1005	
0.1518	-0.0061	0.1005	0.1005	
-0.1598	0.0054	0.1005	0.1005	
0.1669	-0.0047	0.1005	0.1005	
-0.1731	0.0041	0.1005	0.1005	
0.1785	-0.0036	0.1005	0.1005	
-0.1830	0.0031	0.1005	0.1005	
0.1866	-0.0027	0.1005	0.1005	
-0.1894	0.0023	0.1005	0.1005	
0.1913	-0.0020	0.1005	0.1005	
-0.1924	0.0018	0.1005	0.1005	
0.1927	-0.0015	0.1005	0.1005	
-0.1922	0.0013	0.1005	0.1005	
0.1910	-0.0011	0.1005	0.1005	
-0.1891	0.0009	0.1005	0.1005	
0.1865	-0.0008	0.1005	0.1005	

이 또한 0에 가까운 eigenvalue를 얻었고, 검산을 통과했으므로 옳다고 볼 수 있다.

Exercise 4.17.

Inverse iteration method를 반복 실행하여 주어진 행렬의 eigenvalue를 구해보자. 일단 주어진 행렬이 모두 양수인 eigenvalue를 가지고, 4.2.3의 행렬처럼 그 eigenvalue는 0부터 시작하므로, μ 를 0부터 필요한 만큼 키워가면서 반복문을 실행하면 원하는 eigenvalue를 모두얻을 수 있을 것이다. 필요한 MATLAB 구문을 작성하면 다음과 같다.

```
N=input("What are the dimension of the matrix n and max iter?")
max iter=input("What are the dimension of the matrix n and max iter?")
h=1/N; iter=0; tol=1.0e-15; A=zeros(N);
for j=1:N
   A(j,j)=2/h^2;
end
for j=1:N-1
   A(j,j+1) = -1/h^2;
   A(j+1,j) = -1/h^2;
end
A(N,1) = -1/h^2; A(1,N) = -1/h^2;
q=zeros(N,1); q(1)=1; %Initial approximated eigenvector q
prev q=q;
%Using Inverse Iterative Method to find minimum eigenvalue and eigenvector
q=zeros(N,1); q(1)=1; %Initial approximated eigenvector q
prev_q=q;lambda=zeros(41,1);
for j=0:10:400
   T=A-(j+0.1)*eye(N);
   for iter=1:max_iter
       z=T\prev q;
       q=z/norm(z);
       diff_v=norm(q-prev_q);
      prev_q=q;
       if(diff_v<tol)</pre>
          disp("Converged at iter=")
          iter
          break
       end
   lambda(j/10+1)=q'*A*q;
disp("Computed eigenvalue is")
lambda
```

이를 실행하면 다음을 얻는다.

What are the 1 iter = -0.0000 dimension of the -0.0000 matrix n and Converged at iter 211 38.1966 max_iter?10 38.1966 iter = Converged at iter 38.1966
max_iter?10 38.1966 iter = Converged at iter= 38.1966 N = 38.1966 1 iter = 38.1966 10 38.1966 What are the dimension of the iter = 138.1966 dimension of the iter = Converged at iter= 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter= 1 138.1966 500 iter = Converged at iter= 138.1966 Converged at iter= 1 iter = 138.1966 iter = Converged at iter= 1 138.1966 iter = Converged at iter= 1 138.1966 iter = Converged at iter= 1 1 138.1966 1 1 1 138.1966 1 1 1 200 1 1 1 1 201 1 1 1 1 1 201 1 1 1 1 1 1 1 1 1 1 1 1
max_iter?10 38.1966 iter = Converged at iter= 38.1966 N = 38.1966 1 iter = 38.1966 10 38.1966 What are the dimension of the iter = 138.1966 dimension of the iter = Converged at iter= 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter= 1 138.1966 500 iter = Converged at iter= 138.1966 Converged at iter= 1 iter = 138.1966 iter = Converged at iter= 1 138.1966 iter = Converged at iter= 1 138.1966 iter = Converged at iter= 1 1 138.1966 1 1 1 138.1966 1 1 1 200 1 1 1 1 201 1 1 1 1 1 201 1 1 1 1 1 1 1 1 1 1 1 1
iter = Converged at iter= 38.1966 N = 38.1966 1 iter = 38.1966 10 38.1966 Converged at iter= 1 38.1966 What are the dimension of the iter = Converged at iter= 138.1966 matrix n and 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter= 1 138.1966 max_iter = Converged at iter= 1 138.1966 500 iter = Converged at iter= 138.1966 Converged at iter= 1 iter = 138.1966 Converged at iter= 1 iter = 138.1966 iter = Converged at iter= 1 261.8034
N = 38.1966 10 38.1966 10 38.1966 What are the 138.1966 dimension of the iter = Converged at iter = 138.1966 matrix n and 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter = 1 138.1966 500 iter = Converged at iter = 138.1966 Converged at iter = 1 138.1966 Converged at iter = 1 138.1966 iter = Converged at iter = 138.1966 iter = Converged at iter = 138.1966 iter = Converged at iter = 1 261.8034
1 iter = 38.1966 10
Converged at iter= 1 38.1966 What are the dimension of the iter = Converged at iter= 138.1966 matrix n and 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter= 1 138.1966 500 iter = Converged at iter= 138.1966 Converged at iter= 1 iter = 138.1966 Converged at iter= 1 iter = 138.1966 iter = Converged at iter= 1 261.8034
What are the dimension of the iter = Converged at iter = 138.1966 matrix n and 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter = 1 138.1966 max_iter = Converged at iter = 1 138.1966 max_iter = Converged at iter = 1 138.1966 max_iter = Converged at iter = 138.1966 max_iter = 1 138.1966
What are the dimension of the iter = Converged at iter = 138.1966 matrix n and 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter = 1 138.1966 max_iter = Converged at iter = 1 138.1966 max_iter = Converged at iter = 1 138.1966 max_iter = 1 138.
matrix n and 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter= 1 138.1966 500 iter = Converged at iter= 138.1966 Converged at iter= 1 iter = 138.1966 iter = Converged at iter= 1 261.8034
matrix n and 138.1966 max_iter?500 219 iter = 138.1966 max_iter = Converged at iter= 1 138.1966 500 iter = Converged at iter= 138.1966 Converged at iter= 1 iter = 138.1966 iter = Converged at iter= 1 261.8034
Table 138.1966 max_iter = Converged at iter= 1
Table 138.1966 max_iter = Converged at iter
138.1966 500 iter = Converged at iter= 138.1966 138.1966 Converged at iter= 1 iter = 138.1966 138.1966 138.1966 iter = Converged at iter= 1 261.8034
138.1966 500 iter = Converged at iter= 138.1966 138.1966 Converged at iter= 1 iter = 138.1966 138.1966 138.1966 iter = Converged at iter= 1 261.8034
138.1966 Converged at iter= 1 iter = 138.1966 138.1966 iter = Converged at iter= 1 261.8034
138.1966 Converged at iter= 1 iter = 138.1966 138.1966
138.1966 iter = Converged at iter= 1 261.8034
iter = Converged at iter= 1 261.8034
261 0024
201.0034
55 iter = Converged at iter= 261.8034
261.8034
Converged at iter= 1 iter = 261.8034
261.8034
iter = Converged at iter= 1 261.8034
261.8034
144 iter = Converged at iter= 261.8034
261.8034
Converged at iter= 1 iter = 261.8034
361.8034
iter = Converged at iter= 72 361.8034
361.8034
1 iter = C o m p u t e d 361.8034
eigenvalue is 361.8034
Converged at iter= 1 361.8034
lambda = 361.8034
iter = Converged at iter= 400.0000
0.0000 400.0000

따라서 eigenvalue는 0.0000, 38.1966, 138.1966, 261.8034, 361.8034, 400.0000이다. Algebratic multiplicity를 구분하기 위해서는 적당히 위 구문의 vector lambda의 size를 37~45 정도로 조절하면서 확인하면 되는데, 정확히 양 끝값 0, 400을 제외하고는 2개씩 중 근을 갖는 eigenvalue임을 확인할 수 있다.