韓國數學敎育學會誌 시리즈 E
<數學敎育 論文集>
제 22집 제 4호. 2008. 11. 445-457

J. Korea Soc. Math. Ed. Ser. E:
Communications of Mathematical Education
Vol. 22 No 4, Nov. 2008. 445-457

실습을 통한 QR 방법의 효과적인 이해

이 규 봉 (배재대학교)

수치적으로 행렬의 고유치를 모두 계산할 수 있는 방법이 QR 방법이다. 이 방법의 효과적인 교육을 위하여 알고리듬의 이론적인 사실을 실습을 통하여 확인할 수 있는 매트랩 코드와 예를 제시한다.

1. 서 론

과학의 이론이 그 가치를 인정받기 위해서는 일반적으로 그 이론이 옳은지 실험으로 확인해야 한다. 그러나 수학에서는 실험이 별 의미가 없어 대체적으로 이론만 다룬다. 수학 중에서도 수치해석학 분야는 이론으로 개발한 알고리듬이 옳은지 그 결과를 컴퓨터 실습으로 확인할 수 있다. 그러므로 실습을 병행하면 학생들이 알고리듬을 이해하는데 큰 도움을 줄 수 있다.

대학에서 고유치문제를 수치해석적으로 다룰 때 처음 사용하는 방법은 거듭제곱방법(power method)이다. 초기치 벡터에 행렬을 계속하여 거듭 곱해주는 이 방법을 이용하면, 주어진 행렬의 고유치 중 절대값이 가장 큰 고유치와 이에 대응하는 고유벡터를 구할 수 있다. 역행렬에 이 방법을 응용한 방법이 역거듭제곱방법(inverse power method)으로 거듭제곱방법과는 달리 주어진 상수에 가장 가까운 고유치를 구할 수 있다.

거듭제곱방법이나 역거듭제곱방법을 이용하면 단 하나의 고유치와 고유벡터만 구할 수 있다. 이 고유치와 고유벡터를 수축방법(deflation)에 이용하면 주어진 행렬의 다른 고유치와 그에 대응하는 고유벡터를 구할 수 있다. 따라서 이론적으로는 수축방법을 거듭 이용하여 행렬의 모든 고유치와 대응하는 고유벡터를 구할 수 있다. 그러나 행렬이 매우 크면 마무리오차로 인해 정확한 고유치를 구하기에는 효과적이지 못하다.

위에 언급한 방법이 모두 한 번에 단 하나의 고유치를 구하는 것에 반하여, 주어진 행렬을 그와 비슷한 행렬로 변환하여 고유치를 구하는 QR 방법은 한 번에 모든 고유치를 함께 구할 수 있다.

본 논문에서는 대학 수준에서 배우는 QR 방법의 이론적인 결과를 학생들 스스로 그 결과가 옳음을 확인하는 실습과정에 대하여 서술하며, 실습에 필요한 매트랩(matlab) 프로그램과 행렬을 제시한다. 이것을 이용하여 실습을 하면 이론의 단순한 결과 전달에 그치지 않고 학생들 스스로 이론을 이해하

* 2008년 9월 투고, 2008년 9월 심사완료
* ZDM 분류 : N35
* MSC2000 분류 : 97U70
* 주제어 : 고유치, QR 방법, Rayleigh quotient 이동변수, Wilkinson 이동변수

는데 도움을 줄 수 있어 효과적으로 교육할 수 있다. 수축방법에 관한 실습은(이규봉 2006)에서 볼 수 있다.

2장에서는 QR 방법과 이동 QR 방법의 이론적인 관련성과 성질을 설명하며, 3장에서는 실습에 필요한 매트랩 프로그램과 관련 행렬을 제시한다. 그리고 4장에서 마무리한다. 실습에 사용한 매트랩 프로그램은 부록에 따로 정리한다.

2. QR 방법

고유치방정식의 해법은 정사각행렬 A가 주어졌을 때 $Ax=\lambda x$를 만족하는 상수 λ와 벡터 x를 구하는 것이다. 이 상수 λ를 A의 고유치(eigenvalue)라 하고 0이 아닌 벡터 x를 고유치 λ에 대응하는 A의 고유벡터(eigenvector)라고 한다.

정사각행렬 A, B에 대하여 $B=C^{-1}AC$가 되는 정칙행렬 C가 있으면 A, B는 서로 비슷한 행렬(similar matrices)이라고 한다. 비슷한 행렬의 고유치는 서로 같다. 정사각행렬 Q의 역행렬과 전치행렬이 서로 같으면 Q를 직교행렬이라고 한다. 행렬 A에 대하여 $A=QR$이 되는 직교행렬 Q와 위삼각행렬 R로 분해하여 모든 고유치를 한 번에 구할 수 있는 QR 방법은 다음과 같다(이규봉 2003).

알고리듬(QR 방법)

$A^{(0)}=A$

for k=1,2,...

$Q^{(k)}R^{(k)}=A^{(k-1)}$

$A^{(k)}=R^{(k)}Q^{(k)}$

QR 방법의 알고리듬에서 $R^{(k)}=Q^{(k)T}A^{(k-1)}$이므로 $A^{(k)}=Q^{(k)T}A^{(k-1)}Q^{(k)}$이고 따라서 $A^{(k)}=\mathcal{Q}^{(k)^T}A\,\mathcal{Q}^{(k)}$이다. 여기서 $\mathcal{Q}^{(k)}=Q^{(1)}\cdots Q^{(k)}$이다. 직교행렬의 곱은 직교행렬이므로 $\mathcal{Q}^{(k)}$는 직교행렬이다. 따라서 QR 방법으로 중간에 생성되는 행렬 $A^{(k)}$는 모두 A와 비슷한 행렬이다. 그러므로 $A^{(k)}$의 고유치는 모두 A와 같다.

만일 A의 고유치 $\lambda_1,\cdots,\lambda_n$이 $|\lambda_1|>\cdots>|\lambda_n|$을 만족하면 $A^{(k)}$는 고유치가 대각성분 위에 큰 순서대로 놓여 있는 아래와 같은 위삼각행렬로 수렴한다(Johnson).

$$A^{(k)} \to \begin{bmatrix} \lambda_1 & * & * & * \\ & \ddots & * & * \\ & & \ddots & * \\ & & & \lambda_n \end{bmatrix}$$

따라서 A가 대칭행렬이면 $A^{(k)}$도 대칭행렬이므로, 고유치의 절대값의 크기가 다른 대칭행렬은 대각행렬로 수렴한다.

그러나 적당한 $r < n$이 있어 $|\lambda_r| = |\lambda_{r+1}|$이면 $A^{(k)}$는 아래와 같이 블록 위삼각행렬로 수렴하여 r번째 대각성분의 2×2 부분행렬의 고유치가 λ_r, λ_{r+1}이 된다.

$$A^{(k)} \to \begin{bmatrix} \lambda_1 & * & * & * & * & * \\ & \ddots & * & * & * & * \\ & & * & * & * & * \\ & & * & * & * & * \\ & & & & \ddots & * \\ & & & & & \lambda_n \end{bmatrix}$$

그러므로 행렬 A가 복소고유치를 가지면 $A^{(k)}$는 대각성분에 2×2 부분행렬을 갖는 블록 위삼각행렬 형태로 수렴한다. 따라서 블록이 만들어진 2×2 행렬의 고유치는 따로 구한다. 즉 0에 충분히 가깝지 않은 대각성분 바로 아래 항을 찾아 그 항을 2행 1열로 하는 2×2 행렬의 고유치를 따로 구해주어야 한다. 2×2 행렬 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$의 고유치는 다음과 같다.

$$\lambda = \frac{\mathrm{tr}(A) \pm \sqrt{\mathrm{tr}(A)^2 - 4\det(A)}}{2}$$

여기서 $\mathrm{tr}A = a + c, \det(A) = ac - bd$이다.

정리 1. 대칭인 2×2 행렬의 고유치가 서로 같으면 그 행렬은 스칼라행렬이다. 고유치의 절대값은 같지만 부호가 다르면 대각성분도 서로 부호만 다르고 고유치의 제곱은 행렬의 열벡터(또는 행벡터)의 유클리디안 노름의 제곱과 같다.

증명. $A = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$라고 하자. 그러면 이 행렬의 특성방정식은 다음과 같다.

$$\lambda^2 - (a+c)\lambda + ac - b^2 = 0$$

이 방정식이 중근을 가진다고 하자. 그러면 $(a+c)^2 - 4(ac - b^2) = 0$이다. 따라서 $(a-c)^2 + (2b)^2 = 0$이므로 $a = c, b = 0$이다. 따라서 A는 스칼라행렬이다.

A의 고유치를 $\lambda, -\lambda$라고 하자. 그러면 $a+c=0$이고 따라서 $a^2+b^2=\lambda^2$이다.

정리 2. $A=\begin{pmatrix} a & b \\ b & -a \end{pmatrix}$에 QR 방법을 적용하면 중간행렬은 모두 A와 같다.

증명. $A=QR$이라고 하자. 그러면

$$Q=\frac{1}{a^2+b^2}\begin{pmatrix} a & b \\ b & -a \end{pmatrix},\quad R=\begin{pmatrix} \sqrt{a^2+b^2} & 0 \\ 0 & \sqrt{a^2+b^2} \end{pmatrix}$$

이다. $RQ=A$이므로 중간행렬은 모두 같다.

예제 1. $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$의 고유치는 1, -1이고 $\begin{pmatrix} 3 & 4 \\ 4 & -3 \end{pmatrix}$의 고유치는 5, -5이다. 이 행렬에 QR 알고리듬을 적용하면 중간행렬 $A^{(k)}$는 각각 모두 같다.

정리 1과 2에 의하여 A가 대칭행렬이면 $A^{(k)}$는 대각행렬로 수렴하거나, 대각성분에 2×2 부분행렬을 갖는 아래와 같은 삼중대각행렬로 수렴한다.

$$A^{(k)}\rightarrow\begin{bmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & ** & & \\ & & ** & & \\ & & & \ddots & \\ & & & & \lambda_n \end{bmatrix}$$

블록이 만들어진 2×2 행렬의 고유치는 따로 구한다.

행렬 A가 대칭이고 $|\lambda_1|>\cdots>|\lambda_n|$을 만족하면 $A^{(k)}$는 $D=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$에 수렴하며 그 속도는 인접한 고유치의 비로 결정된다. 즉 $C=\max|\lambda_i|/|\lambda_{i-1}|$이면

$$\|A^{(k)}-D\|=O(C^k)$$

이다.

A를 대칭행렬이라고 하자. 그러면 고유치는 모두 실수이다. QR 방법의 각 단계마다 A의 고유치에 가까운 상수만큼 변화를 주는 다음과 같은 방법을 이동 QR 방법이라 한다.

알고리듬(이동 QR 방법)

$A^{(0)} = A$

for k=1,2,...

상수 $\mu^{(k)}$ 선택

$Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I$

$A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$

end

이동 QR 방법으로 중간에 생성되는 행렬 $A^{(k)}$도 모두 A와 비슷한 행렬이다. 그러므로 $A^{(k)}$의 고유치는 모두 A와 같다. 이동변수 $\mu^{(k)}$는 반복과 관계없는 일정한 상수로 택할 수도 있다. 만일 $\mu^{(k)} = 0$으로 택하면 QR 방법이 된다.

이동변수 $\mu^{(k)}$를 $\underline{Q}^{(k)} = [q_1^{(k)}, \cdots, q_n^{(k)}]$의 마지막 열벡터에 대응하는 Rayleigh Quotient $\mu^{(k+1)} = q_n^{(k)T} A q_n^{(k)}$로 사용하자. 그러면

$$A_{nn}^{(k)} = e_n^T A^{(k)} e_n = e_n^T \underline{Q}^{(k)^T} A \underline{Q}^{(k)} e_n = q_n^{(k)T} A q_n^{(k)}$$

이므로 $\mu^{(k+1)} = A_{nn}^{(k)}$이다. 즉 $A^{(k)}$의 마지막 성분이다. 모든 초기 조건에 대하여 수렴하는 것은 아니지만 일반적으로 Rayleigh quotient 이동변수를 사용하면 빨리 수렴한다(Trefethen).

이동변수로 Wilkinson의 방법을 사용할 수 있다. B를 $A^{(k)}$의 맨 마지막에 있는 2×2 행렬이라 하고 성분을 아래와 같다고 하자.

$$B = \begin{pmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{pmatrix}$$

Wilkinson 이동변수 μ는 a_n에 가까운 B의 고유치로 정한다. 즉

$$\mu = a_n - \operatorname{sign}(\delta) b_{n-1}^2 / (|\delta| + \sqrt{\delta^2 + b_{n-1}^2})$$

여기서 $\delta = (a_{m-1} - a_m)/2$이다. 만일 $\delta = 0$이면 $\operatorname{sign}(\delta) = \pm 1$ 어느 것이든 택한다. B의 고유치 둘 다 같으면 정리에 의하여 $b_{n-1} = 0$이므로 $\mu = a_n$이 된다.

대칭행렬인 경우 이동 QR 방법은 일반적으로 대각행렬로 수렴한다. 그러나 Rayleigh quotient 이동변수를 사용하는 경우 수렴하지 않는 경우도 있다. 그러나 Wilkinson 이동변수를 사용하면 대각행렬로 수렴한다(Trefethen).

예제 2. $A=\begin{pmatrix}0&1\\1&0\end{pmatrix}$, $B=\begin{pmatrix}3&4\\4&-3\end{pmatrix}$

(1) 위 행렬 A, B 모두 QR 방법을 이용하면 대각행렬로 수렴하지 않는다.

(2) B에 이동 QR 방법을 이용하면 대각행렬로 수렴한다.

(3) A에 Rayleigh quotient 이동변수를 사용하면 대각행렬로 수렴하지 않는다. 그러나 Wilkinson 이동변수를 사용하면 수렴한다.

3. 실습을 통한 효과적인 이해

이 장에서는 2장에서 제시된 아래와 같은 이론적 결과가 옳은지 실습으로 확인한다. 이를 위하여 매트랩을 이용한 프로그램과 예제로 사용될 행렬을 제시한다. 프로그램 코드는 부록에 있다.

[1] $A^{(k)}$의 고유치는 모두 A와 같다.
[2] A가 대칭행렬이면 $A^{(k)}$도 대칭행렬이다.
[3] $A^{(k)}$는 대각성분에 2×2 부분행렬을 갖는 블록 위삼각행렬 형태로 수렴한다.
[4] A의 고유치가 $|\lambda_1|>\cdots>|\lambda_n|$을 만족하면 $A^{(k)}$는 고유치가 크기순으로 대각성분 위에 배열되는 위삼각행렬로 수렴한다. 단 A가 블록행렬이면 블록 단위로 수렴
[5] 대칭행렬에 Wilkinson 이동변수를 사용하면 대각행렬로 수렴한다.

[1]을 실습하기 위하여 매트랩 프로그램 gen_MAT(m,n,k,p)과 alg_QR_Ak(A,kmax)이 필요하다. gen_MAT(m,n,k,p)은 크기가 $m\times n$이고 성분이 $[-k,k]$ 사이에 있는 정수로 이루어진 임의의 행렬을 출력한다. 여기서 p=0이면 일반행렬, p=1이면 대칭행렬, p가 그 외의 수이면 강대각지배행렬을 출력한다. alg_QR_Ak(A,kmax)은 행렬 A에 QR 방법을 kmax 번 반복한 결과 $A^{(\mathrm{kmax})}$를 출력한다.

실습 1. alg_QR_Ak(A,kmax)과 gen_MAT(m,n,k,p)을 반복 사용하여 [1]를 입증한다.

```
>> A=gen_MAT(4,4,5,0);
>> A_k=alg_QR_Ak(A,10);
>> [eig(A)'; eig(A_k)']
ans =
-4.1141        1.1559 - 4.1504i   1.1559 + 4.1504i   2.8024
-4.1141        1.1559 - 4.1504i   1.1559 + 4.1504i   2.8024
```

다양한 변수 kmax, n, k, p를 사용하여 [1]을 입증한다. 출력된 고유치의 순서가 바뀔 수 있으나

두 행렬의 고유치는 모두 같음을 알 수 있다. 그 결과 QR 방법에 의해 중간에 만들어진 $A^{(k)}$의 고유치는 이론대로 모두 A와 같음을 학생은 확인할 수 있다.

실습 2. alg_QR_Ak(A,kmax)과 gen_MAT(m,n,k,p)을 반복 사용하여 [2]를 입증한다. 대칭행렬이므로 p=1을 사용한다.

```
>> A=gen_MAT(5,5,10,1);
>> A_k=alg_QR_Ak(A,50)
A_k =
   -9.5603    0.0000    0.0000    0.0000    0.0000
    0.0000   -5.8664    0.2186    0.0000    0.0000
    0.0000    0.2186    5.4725    0.0003    0.0000
    0.0000    0.0000    0.0003   -4.4680   -0.0000
   -0.0000   -0.0000   -0.0000    0.0000    0.4223
```

다양한 변수 kmax, m=n, k를 사용하여 대칭행렬이면 중간에 형성되는 행렬 $A^{(k)}$도 모두 대칭행렬이 됨을 확인할 수 있다.

실습 3. alg_QR_Ak(A,kmax)과 gen_MAT(m,n,k,p)을 반복 사용하여 [3]를 입증한다.

(1) 다음 행렬을 대각성분으로 갖는 블록위삼각행렬을 만든다. 행렬 A의 고유치가 1,4,5이고 B의 고유치가 5, -5이므로 블록위삼각행렬의 고유치는 1, 4, 5, 5, -5가 된다.

$$A=\begin{pmatrix} 2 & 1 & -1 \\ 1 & 4 & 1 \\ -1 & 1 & 4 \end{pmatrix},\quad B=\begin{pmatrix} 3 & 4 \\ 4 & -3 \end{pmatrix}$$

```
>> A=[2 1 -1;1 4 1;-1 1 4]; B=[3 4;4 -3];
>> C=[A gen_MAT(3,2,5,0); zeros(2,3) B];
>> alg_QR_Ak(C,30)
ans =
    4.0000    0.0000   -0.0000    1.7321    0.0000
    0.0000    5.0000   -0.0000   -0.7071    2.8284
   -0.0000    0.0000    1.0000    1.2247    2.4495
         0         0         0    3.0000    4.0000
         0         0         0    4.0000   -3.0000
```

(2) 복소고유치 경우

```
>> A=gen_MAT(5,5,10,0);
>> eig(A)'
ans = -17.1188    -9.5631    2.4247 + 3.1317i   2.4247 - 3.1317i   7.8325
>> A_k=alg_QR_Ak(A,100)
A_k =
  -17.1188   -2.1559   11.6364   -7.6189   -0.2486
    0.0000   -9.5631   -6.4451   -2.8208   -7.1722
    0.0000   -0.0000    7.8325   -3.6454    3.2436
   -0.0000   -0.0000    0.0000   -0.1507   13.5177
   -0.0000   -0.0000   -0.0000   -1.2162    5.0002
>> eig(A_k(4:5,4:5))'
ans =
   2.4247 + 3.1317i   2.4247 - 3.1317i
```

kmax를 10, 30, 50, 100과 같이 점차 크게 하면 A_k가 블록 위삼각행렬 형태로 수렴함을 알 수 있다. A의 고유치는 대각성분에 놓여 있고 나머지는 블록 2×2 부분행렬의 고유치임을 확인할 수 있다.

실습 4. gen_MAT(m,n,k,1)을 반복 사용하여 서로 다른 고유치가 나오는 행렬을 만든 후 alg_QR_Ak(A,kmax)을 사용하여 [4]를 입증한다.

(1) 크기순으로 대각성분 위에 놓인다.

```
>> A=gen_MAT(5,5,10,1);
>> eig(A)'
ans =
  -12.2691   -6.9111   -0.8371    4.0890   11.9283
>> A_k=alg_QR_Ak(A,100)
A_k =
  -11.3482    4.6299    0.0000   -0.0000    0.0000
    4.6299   11.0074    0.0000    0.0000   -0.0000
   -0.0000   -0.0000   -6.9111   -0.0000   -0.0000
    0.0000   -0.0000   -0.0000    4.0890   -0.0000
    0.0000    0.0000    0.0000   -0.0000   -0.8371
```

```
>> A_k=alg_QR_Ak(A,400)
A_k =
  -12.2691    0.0010    0.0000   -0.0000    0.0000
    0.0010   11.9283    0.0000    0.0000   -0.0000
   -0.0000   -0.0000   -6.9111   -0.0000   -0.0000
    0.0000   -0.0000   -0.0000    4.0890   -0.0000
         0         0         0   -0.0000   -0.8371
```

kmax를 100, 200, 300, 400과 같이 점차 크게 하면 결국 이론대로 크기순으로 수렴하는 것을 확인할 수 있다.

(2) A가 블록행렬이면 블록 단위로 고유치가 수렴한다.

```
>> B=[4 1 1 2 3;0 6 3 1 2;-2 1 4 3 2; 0 0 0 4 5;0 0 0 3 6];
>> B_k=alg_QR_Ak(B,50)
B_k =
    6.0002    3.2659   -0.3651   -3.4786   -0.9486
   -0.0001    4.9998    1.7889   -2.3236    0.7746
   -0.0000   -0.0000    3.0000    3.4641   -0.0000
         0         0         0    9.0000    2.0000
         0         0         0    0.0000    1.0000
```

kmax를 10, 30, 50, 70, 100과 같이 점차 크게 하면 B는 블록행렬이므로 블록 단위별 크기 순으로 고유치가 대각성분에 수렴하는 것을 확인할 수 있다. 실습 3의 (1)에서도 kmax를 점차 크게 하면 (kmax=250) 결국 고유치가 큰 순서로 배열됨을 알 수 있다.

(3) 행렬 A가 복소고유치를 가지면 $A^{(k)}$는 대각성분에 2×2 부분행렬을 갖고, 이 부분행렬의 고유치가 A의 고유치가 된다.

```
>> A=[-2 -2 -9;-1 1 -3; 1 1 4];
>> eig(A)'
ans =
   1.0000 - 1.0000i   1.0000 + 1.0000i   1.0000
>> A_k=alg_QR_Ak(A,30)
A_k =
    0.3637   -6.3095   -8.6162
    0.2227    1.6363   -0.3016
```

```
   -0.0000    0.0000    1.0000
>> eig(A_k(1:2,1:2))'
ans =
   1.0000 + 1.0000i   1.0000 - 1.0000i
```

(4) 다음 대칭행렬의 고유치는 고유치의 크기가 모두 다름에도 불구하고 대각행렬로 수렴하는 것 같지 않다. 그러나 kmax를 충분히 크게 하면(kmax=5000) 결국 대각행렬로 수렴함을 알 수 있다.

$$A=\begin{pmatrix} -5 & 7 & 0 & -3 & -1 & 6 & 7 & 1 \\ 7 & -7 & -3 & -4 & 0 & 5 & 6 & -4 \\ 0 & -3 & 1 & -2 & 5 & -7 & 3 & 7 \\ -3 & -4 & -2 & 3 & 1 & -3 & -8 & 1 \\ -1 & 0 & 5 & 1 & -10 & -7 & 0 & 3 \\ 6 & 5 & -7 & -3 & -7 & -5 & -1 & 3 \\ 7 & 6 & 3 & -8 & 0 & -1 & -4 & 7 \\ 1 & -4 & 7 & 1 & 3 & 3 & 7 & -8 \end{pmatrix}$$

[5]를 실습하기 위하여 alg_QR_RQ_Ak(A,kmax)와 alg_QR_WK_Ak(A,kmax)을 제시한다. 전자는 행렬 A에 Rayleigh quotient 이동 QR 방법을, 후자는 행렬 A에 Wilkinson 이동 QR 방법을 kamx 번 반복한 결과 $A^{(kmax)}$를 출력한다.

실습 5. 대칭행렬에 Wilkinson 이동변수를 사용하면 대각행렬로 수렴한다.

```
>> A=[0 1;1 0]; B=[3 4; 4 -3];
>> alg_QR_Ak(A,5)          % 또는 alg_QR_Ak(A,5) 모두 QR 방법 수렴하지 않음
ans =
     0     1
     1     0
>> alg_QR_RQ_Ak(B,5)    % 또는 alg_QR_WK_Ak(B,5) 이동 QR 방법 수렴
ans =
    5.0000   -0.0000
    0.0000   -5.0000
>> alg_QR_RQ_Ak(A,5)    % Rayleigh quotient 이동 QR 방법 수렴하지 않음
ans =
     0     1
     1     0
>> alg_QR_WK_Ak(A,5)   % Wilkinson 이동 QR 방법 수렴
```

```
ans =
   1.0000    0.0000
        0   -1.0000
>> alg_QR_RQ_Ak(C,30)  % 실습 3의 행렬 C도 수렴한다.
ans =
   4.0000    0.0000   -0.0000    1.5492   -0.7746
   0.0000    5.0000    0.0000    0.6325    2.8460
  -0.0000    0.0000    1.0000    2.1909    1.6432
        0         0         0    5.0000    0.0000
        0         0         0         0   -5.0000
```

4. 결 론

일반적으로 수학에서 실습은 중요하지 않으나 수치해석학은 실습이 필요한 과목이다. 수학적으로 증명된 알고리듬과 그 성질을 컴퓨터를 이용하여 실습하고 그 결과가 옳음을 확인하는 것은 매우 교육적이며 바람직하다.

실습을 통해서 QR 방법의 이론적 결과가 옳음을 보여주었다. 그러나 가끔 이론대로 나오지 않을 수도 있다. 이런 경우 왜 그런지 파악하여야 한다. 예를 들어, QR 방법은 가장 큰 고유치 순서로 대각행렬에 수렴하게 되어 있으나 충분한 반복을 하지 않으면 순서가 달라질 수 있다. 이 경우 충분히 더 반복하면 예상한대로 가장 큰 고유치에 수렴함을 보일 수 있다. 또한 고유치 간격의 비가 작을수록 또는 고유치가 멀리 떨어져 있을수록 빨리 수렴하는 것도 보일 수 있다.

본 논문에서는 내용을 대학 수준으로 제한하였으나, 보다 실제적인 계산에서는 복잡도를 줄이기 위하여 행렬을 먼저 비슷한 삼중대각행렬로 변화시킨 후 QR 방법에 적용한다(Trefethen).

이론과 실습이 서로 조화를 이루어 수치해석학을 교육하면 이론을 이해하는데 도움이 되며, 수학의 중요성을 더욱 강하게 학생들에게 보여줄 수 있다.

참 고 문 헌

이규봉 (2003). 실습하며 배우는 수치해석학, 서울: 경문사

이규봉 (2006). 실습을 통한 수축방법의 효과적인 이해, 한국수학교육학회 시리즈 E <수학교육 논문집> **20(4)**, pp.575-586.

Johnson, L. W. & Riess, R. D. (1982). *Numerical Analysis*, Addison-Wesley

Trefethen, L. N. & Bau, D. (1997). *Numerical Linear Algebra*, SIAM, Philadelphia

Effective Teaching of QR methods using Computer Practice

Gyou Bong Lee
Dept. of Applied Math., Paichai University, Daejeon 302-175, Korea
E-mail: gblee@pcu.ac.kr

Although both theory and experiment are very important parts in sciences, especially in mathematics, theory only seems to be very important. But the subject of numerical analysis needs both theory and practice in computer. In this paper, I provide some Matlab program codes and matrices which are used in good understanding QR methods in class.

* ZDM Classification : N35
* 2000 Mathematics Subject Classification : 97U70
* Key Words : eigenvalue, QR method, Rayleigh quotient shift, Wilkinson shift

<부록>

```
function A=gen_MAT(m,n,k,p)
A=round(2*k*rand(m,n)-k);
if p==0
    return
end
if m~=n
    error('m과 n이 다름')
end
A=round((A+A')/2);
if p==1
    return
end
for i=1:n
    A(i,i)=sum(abs(A(i,:)));
end

function A_k=alg_QR_Ak(A,kmax)
for k=1: kmax
    [Q,R]=qr(A);
    A=R*Q; % 중간행렬
end
A_k=A;

function [A_k,lambda]=alg_QR_RQ_Ak(A,kmax)
n=length(A);
for k=1: kmax
    mu=A(n,n); % Rayleigh quotient 이동변수
    [Q,R]=qr(A-mu*eye(n));
    A=R*Q+mu*eye(n);
end
A_k=A;

function A_k=alg_QR_WK_Ak(A,kmax)
n=length(A);
for k=1: kmax
    del=(A(n-1,n-1)-A(n,n))/2;
    if del>=0
        sign_del=1;
    else
        sign_del=-1;
    end % Wilkinson 이동변수
    mu=A(n,n)-sign_del*A(n,n-1)^2/(abs(del)+sqrt(del^2+A(n,n-1)^2));
    [Q,R]=qr(A-mu*eye(n));
    A=R*Q+mu*eye(n);
end
A_k=A;
```