Numerical Methods

**Assignment4: Finding Eigenvalue using QR Iteration method**

**Name: Hong Se Hyun**  **ID: 21700791**

**Question. Condition Number**

Determine the condition number of matrix A and explain whether this is ill-conditioned system. Use 2-norm to calculate the condition number:

**or**

**where is pseudoinverse**

Program the function of

***double cond(matrix A)***

* A: (mxn) can be a rectangular matrix
* Need to display the condition number output
* Find the condition number for

**A =**  **C =**

**Procedure**

The condition number for non-singular matrix is and singular matrix, it is where is the pseudoinverse.

The 2-norm of matrix **A** is defined as , where **)**.

From using the relationship of

the condition number can be calculated as

**,** i=1 to r (rank r) ()

**1. Create C++ function of eig()**

* Write a pseudocode for finding eigenvalue with iteration and QR decomposition

Double eig(A)

for i = 1 to N // or until becomes *U*

***// Part 1. QRdecomp using household matrix***

[ Q R ] = QRDecomp();

***// Part 2. Make into similar matrix***

*end*

*U =*

*λ = diagElement(U)*

*return λ*

* Write a pseudocode for [Q R] = QRDecomp() // see Appendix

R = A v = c + e

Q = I u = v /

For k = 0 to n-1 H = H -

H = I // use ‘eye’ function

c, v, e = 0 Q = QH

c = R ( i : n , k) R = HR

end

if c(k) > 0

e(k) = 1 end function

else

e(k) = -1

end

* Create C++ functions of eig( )

|  |  |
| --- | --- |
|  |  |
| Figure 1.1 Functions of eig | Figure 1.2 Functions of eig |
|  |  |
| Figure 1.3 Functions of eig | Figure 1.4 Functions of eig |

* Need to check if the input matrix is a square matrix
* Need to check if there are any division by zero

|  |
| --- |
|  |
| Figure 2.1 Error Checking(1) |
|  |
| Figure 2.2 Error Checking(2) |

*A*가 *Square Matrix*인지 아닌지를 판별해주는 조건문을 함수 제일 앞에 배치함으로써, 행렬을 받아오자마자 정방행렬 여부를 가장 먼저 따져주도록 코드를 작성하였다.

한편, 최종적으로 *Condition Number*를 계산해주는 과정 에서 이 되어야 *Condition Number*가 무한대로 발산하지 않고 정상적인 값을 얻어낼 수 있다. 즉, 인 경우에는 *Error*라고 사용자에게 알려주기 위해, 함수의 끝부분에서 값을 얻어낸 뒤에 Figure 2.2와 같은 조건문을 만들었다.

**2. Get eigenvaules of ()**

myMatrix.cpp 파일에 만들어준 *transpose*함수와 *multiply\_Mat*함수를 이용하여 를 바로 얻을 수 있다.

|  |  |
| --- | --- |
|  |  |
| Figure 3.1 Function of Transpose | Figure 3.2 Function of multiply\_Mat |

|  |
| --- |
|  |
| Figure 3.3 Main Code of Finding and |

와 를 구한 뒤에, Figure 1.1 ~ Figure 1.4에 있는 함수를 이용하여 eigenvalue를 얻을 수 있다.

|  |  |
| --- | --- |
|  | |
| Figure 3.4 Main Code of Calculate Eigenvalues of and | |
|  |  |
| Figure 3.5 Result of and | Figure 3.6 Eigenvalues of and |

**3. Find the maximum and minimum eigenvalues**

myMatrix.cpp 파일에 만들어준 *max*함수와 *min*함수를 이용하여 의 maximum eigenvalue와 minimum eigenvalue, 그리고 의 maximum eigenvalue와 minimum eigenvalue를 바로 얻을 수 있다

|  |  |
| --- | --- |
|  |  |
| Figure 4.1 Function of Finding Max | Figure 4.2 Function of Finding Min |
|  | |
| Figure 4.3 Main Code of Finding Max Eigenvalue and Min Eigenvalue | |
|  | |
| Figure 4.4 Result of Max and Min Eigenvalue | |

**4. Square-root the ratio of max-min values.**

위 식을 이용하여 와 의 *Condition Number를 바로 찾아줄 수 있다.*

|  |  |
| --- | --- |
|  |  |
| Figure 5.1 Function of condNumber | Figure 5.2 Function of condNumber |
|  | |
| Figure 5.3 Main Code of Finding Condition Number | |
|  | |
| Figure 5.4 Result of Condition Number | |

Condition Number에 대한 정의는 이지만,

**,** *i* =1 to r (rank r) ()

를 통해서도 동일한 Condition Number를 얻을 수 있다는 개념에 입각하여 Condition Number를 얻을 수 있었다.