Numerical Analysis
homework 07: Matrix Eigenvalues

Due on Tuesday, April 18, 2017

102061149 Fu-En Wang

1 Introduction

In previous homework, we had already use power method and inverse power method to find the largest and smallest eigenvalue. However, both of them cannot be used to find all the eigenvalue of a matrix A; therefore, it this project, we will use **QR Method** to find all the eigenvalues.

1.1 API

In **MAT.h** and **MAT.cpp**, I implement the following functions:

1. **void QRFact(const MAT &A, MAT &Q, MAT &R);**
2. **int EVqr(MAT &A, double tol, int maxiter);**
3. **int EVqrShifted(MAT &A, double mu, double tol, int maxiter);**

QRFact will apply **QR Decomposition** to A and store them into Q and R. **EVqr** and **EVqrShifted** is the implementation of **QR Iteration** and **Shifted QR Iteration**. In this assignment, **tol** should be set to 10^{-9} and **mu** should be 0.5. We have to find the three largest and the three smallest eigenvalues of **m3.dat**, **m4.dat**, **m5.dat**, **m6.dat**, **m7.dat**, **m8.dat**.

1.2 Error Calculation

According to **Gram-Schmidt Process**, we should get a final matrix with 0 in the non-diagonal from **QR Iteration**. Therefore, it makes sense that we can check whether all non-diagonal elements are zero or not. However, for saving execution time, we simply this checking to the following fomula:

$$error = \max_{2 \le i \le n} |a_{i,i-1}|$$

For each iteration, we only check the one-line elements below diagonal, which will be faster a lot than checking all non-diagonal elements.

2 Implementation

Algorithm 1 QR Decomposition

$A = \{a_1, a_2, ..., a_n\}, a_i$ is column vector,
$r_{11} = \sqrt{(a_1)^T a_1}$
$q_1 = \frac{a_1}{r_{11}}$
for each j $\in$ {2, ..., n} **do**
  $q_j = a_j$
  for each i $\in$ {1, ..., j-1} **do**
    $r_{ij} = (q_i)^T q_j$
    $q_j -= r_{ij} q_i$
  end for
  $r_{jj} = \sqrt{(q_j)^T q_j}$
  $q_j = \sqrt{sum((q_j)^2)}$
end for

Algorithm 2 QR Iteration

$T^{(0)} = A$
for each k ∈ {1, ..., maxIter} **do**
  $T^k = Q^k R^k$
  $T^{k+1} = R^k Q^k$
  if error < tol **then**
    break
  end if
end for

Algorithm 3 Shifted QR Iteration

$T^{(0)} = A$
for each k ∈ {1, ..., maxIter} **do**
  $T^k - \mu I = Q^k R^k$
  $T^{k+1} = R^k Q^k + \mu I$
  if error < tol **then**
    break
  end if
end for

2.1 Complexity

For **QR Decomposition**, the double for-loop do $\frac{n(n+1)}{2}$ times of operation, and $(q_i)^T q_j$ in each operation make **QR Decomposition** be a $\boldsymbol{O(n^3)}$ problem.
For each iteration of **QR Iteration**, there are **QR Decomposition** and **mat** $\times$ **mat**. So it is also $\boldsymbol{O(n^3)}$ problem.