YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CptSci 440/540: Numerical Computation I | Handout # | April 14, 2016

V. Rokhlin

Problem Set 3

Due Date: April 29

In this problem set, you will implement the QR algorithm with shifts to find the spectrum of a user-provided symmetric tridiagonal matrix.

The algorithm has been discussed in class. It goes as follows (for the sake of clarity, I explain it for the case $n = 4$).

Step 1. You are given a symmetric TRIDIAGONAL $n \times n$ matrix A. You copy it to a matrix B.

Step 2. You perform the shift:

$$\tilde{B} = B - \mu I, \tag{1}$$

where $\mu = B(1,1)$ is the value in the upper left corner.

Step 3. You bring $\tilde{B}$ to the lower triangular form. To do so, you apply $n-1$ Givens rotation to the rows of $\tilde{B}$, starting from the bottom and going up. In other words, the order of operations is

$$\tilde{B} = \begin{pmatrix} \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot \end{pmatrix} \overrightarrow{Q_3} \begin{pmatrix} \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & 0 \\ 0 & * & * & 0 \\ 0 & * & * & * \end{pmatrix} \overrightarrow{Q_2} \begin{pmatrix} \cdot & \cdot & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & \cdot & \cdot & \cdot \end{pmatrix} \overrightarrow{Q_1} \begin{pmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot \end{pmatrix} = L, \tag{2}$$

where Q_j is a Givens rotation that combines the rows j and $j+1$ to put a zero at the position $(j, j+1)$.

Step 4. You apply the adjoints of the Givens rotations from the previous step to L, that is

$$L \to L \cdot Q_3^T \cdot Q_2^T \cdot Q_1^T. \tag{3}$$

Note that the resulting matrix is again tridiagonal and symmetric.

Step 5. You add the shift μ back to the diagonal.

To summarize, the basic iteration of the algorithm transforms B into B_{new} via

$$B \to B - \mu I \to Q(B - \mu I) = L \to LQ^T = Q(B - \mu I)Q^T \to LQ^T + \mu I = QBQ^T = B_{new}, \tag{4}$$

where $Q = Q_1 \cdot Q_2 \cdot Q_3$.

You keep on iterating Steps 2-5 until the value at the position $(1,2)$ (just next to the upper left corner) is very small ($B_{new}(1,2) \sim 0$). Then you declare the value in the upper left corner to be your approximation of the first eigenvalue of A (why does it make sense?), and apply the same iterations to the $(n-1) \times (n-1)$ submatrix obtained from B_{new} by deleting the first column and the first row. You continue until you have found all the eigenvalues of A.

You need to write a code that implements the algorithm described above.

In FORTRAN, your calling sequence should be

$$qr\_symmetric(a, n, b) \tag{5}$$

where

$a(n,n)$ is a (real) $n \times n$-matrix to be diagonalized (input parameter). Assume that a is a SYMMETRIC TRIDIAGONAL matrix.

n is the (integer) size of the matrix (input parameter)

$b(n,n)$ is a (real) $n \times n$-matrix (output parameter). The diagonal of b (i.e. $b(1,1), b(2,2), \ldots$) will contain the spectrum of a.

In C, your calling sequence should be

$$void\ qr\_symmetric(double *\ a, int\ n, double *\ b), \tag{6}$$

where

a points to an array of doubles of size n^2, containing $a(1,1), a(1,2), \ldots, a(1,n), a(2,1), \ldots, a(n,n)$, a being the $n \times n$ matrix to be diagonalized (input parameter). Assume that a is a SYMMETRIC TRIDIAGONAL matrix.

n is the (integer) size of the matrix (input parameter)

b points to an array of doubles of size n^2, containing $b(1,1), b(1,2), \ldots, b(1,n), b(2,1), \ldots, b(n,n)$, b being the $n \times n$ matrix. The diagonal of b $(b(1,1), b(2,2), \ldots)$ will contain the spectrum of a.