# Homework 4 Report

Anthony Falcon

**To run the code run `make -f Makefile`.**

For the coding portion of this homework, we were asked to create four subroutines to complete the following tasks: Reducing a symmetric matrix to tridiagonal, Calculates the eigenvalues using the QR algorithm with shifts and without shifts and finally create a program to calculate the eigenvector of a given eigenvalue.

For the first problem we used the householder method to tridiagonalize a symmetric matrix using the algorithm on pg.114 of the lecture notes (pg. 24 in the chapter 4 lecture notes). This algorithm uses householder matrices similar to the QR factorization to compute the tridiagonalization. The input matrix is:

| | | | |
|---|---|---|---|
| 5 | 4 | 1 | 1 |
| 4 | 5 | 1 | 1 |
| 1 | 1 | 4 | 2 |
| 1 | 1 | 2 | 4 |

After running the algorithm on the above matrix, we get out the following tridiagonal matrix.

| | | | |
|---|---|---|---|
| 5.000 | -4.243 | 0.000 | 0.000 |
| -4.243 | 6.000 | 1.414 | 0.000 |
| 0.000 | 1.414 | 5.000 | 0.000 |
| 0.000 | 0.000 | 0.000 | 2.000 |

Full output with higher precision is in output.txt.

For the second problem we needed to program two different algorithms to find the eigenvalues of a symmetric matrix. The first one was QR algorithm without shifts where we simply computed the QR factorization and then preformed the matrix multiplication RQ. This returned to us a diagonal matrix of the eigenvalues. The second algorithm was QR factorization with shifts and deflation. Similar to the first eigenvalues finding algorithm this subtracts off from the diagonal the A(m,m) entry from A (the shift). Since this finds the last eigenvalue quickly we need to then reduce the matrix we are working on by removing the last row and column of the matrix rerunning the algorithm on it. This will then return the eigen values of the matrix.
To test the code we ran it on this matrix:

| | | |
|---|---|---|
| 3 | 1 | 0 |
| 1 | 2 | 1 |

| | | |
|---|---|---|
| 0 | 1 | 1 |

And as output we got the following for QR without shifts:

| |
|---|
| 3.7321 |
| 2.0000 |
| 0.2679 |

For QR with shifts and deflation:

| |
|---|
| 3.7321 |
| 1.9999 |
| 0.2679 |

Once again higher precision is available in output.txt.

Finally, the last algorithm we programmed was to find eigenvectors when we are given approximate values for eigenvalues following the algorithm on page 98 of the lecture notes (pg. 8 of chapter 4 lecture notes). To test our code we ran it on the following matrix.

| | | | |
|---|---|---|---|
| 2 | 1 | 3 | 4 |
| 1 | -3 | 1 | 5 |
| 3 | 1 | 6 | -2 |
| 4 | 5 | -2 | -1 |

For the output matrix we got:

| | | | |
|---|---|---|---|
| 0.26346 | 0.56014 | 0.37870 | 0.68805 |
| 0.65904 | 0.21163 | 0.36242 | -0.62412 |
| -0.19963 | 0.77671 | -0.53794 | -0.25980 |
| -0.67557 | 0.19538 | 0.66020 | -0.26375 |

Again the matrix with double precision is available in output.txt.