CS 475/CM 375 - Fall 2010: Assignment 3

Instructor: Pascal Poupart Office: DC2514 Email: ppoupart@uwaterloo.ca
Classroom: MC 4060 TuTh 10-11:20am Office hours: TuTh 2:30-4pm
Class homepage: www.student.cs.uwaterloo.ca/˜cs475/

Due: November 18, Thursday (in class)

1. (6 marks) You are given vectors x and y. Let $Q = I - 2\frac{vv^T}{v^Tv}$ be a Householder matrix. Find v such that Qx is a multiple of y; i.e. $Qx \in \text{span}\{y\}$.

2. (3 marks) A nonzero vector y is called a left eigenvector if $y^T A = \mu y^T$. Let λ and μ be distinct eigenvalues of A, let x be an eigenvector for λ, and let y be a left eigenvector for μ. Show that x and y are orthogonal.

3. (15 marks) Let A be a symmetric tridiagonal matrix.
   (a) In the QR factorization of $A = QR$, which entries of R are in general nonzero? Which entries of Q? Explain your answer.
   (b) Show that the tridiagonal structure is recovered when the product RQ is formed. (Hint: Show that (i) RQ is upper Hessenberg, and (ii) RQ is symmetric.)
   (c) Explain how (2×2) Householder transformation can be used in the computation of the QR factorization of a tridiagonal matrix. Estimate the complexity of your algorithm.

4. (15 marks) Suppose A is an $n \times n$ symmetric positive definite matrix. Consider the following iteration:

$$
\begin{aligned}
& A_0 = A \\
& \text{for } k = 1, 2, \ldots \\
& \qquad A_{k-1} = G_k G_k^T \qquad \text{(Cholesky factorization)} \\
& \qquad A_k = G_k^T G_k \\
& \text{end}
\end{aligned}
$$

   (a) Show that if A_{k-1} is symmetric positive definite, so is A_k. Also show that A_{k-1} and A_k have the same eigenvalues.
   (b) Suppose $(n = 2)$

$$
A_{k-1} = \begin{bmatrix} a_{k-1} & b_{k-1} \\ b_{k-1} & c_{k-1} \end{bmatrix}, \qquad A_k = \begin{bmatrix} a_k & b_k \\ b_k & c_k \end{bmatrix}.
$$

   $(b_{k-1} \neq 0.)$ Compute a_k, b_k, and c_k in terms of a_{k-1}, b_{k-1} and c_{k-1}. (i.e. perform one iteration of the algorithm)
   (c) From part (b), we know that $b_k = b_{k-1}\alpha_{k-1}$ where

$$
\alpha_{k-1} = \frac{\sqrt{a_{k-1}c_{k-1} - b_{k-1}^2}}{a_{k-1}}.
$$

   Show that $\alpha_k \leq \alpha_{k-1}$.

(d) Using part (c), show that $b_k \to 0$ if $a_0 \geq c_0$. (Hint: First show that $b_k \leq b_0 \alpha_0^k$, and then show that $\alpha_0 < 1$ using the fact that $a_0 c_0 > b_0^2$ since A_0 is symmetric positive definite.)

(e) Using part (d), show that $A_k \to \text{diag}(\lambda_1, \lambda_2)$, where λ_1 and λ_2 are eigenvalues of A.

5. (8 marks) Implement the Householder QR factorization for symmetric, band matrices with bandwidth p. Create a MATLAB function:

```
[Q,R] = BandQR(A,p)
```

The inputs are the matrix A and bandwidth p, and the outputs are the Q, R factors of A. The algorithm should not operate on the zeros of A. To start, try $p = 3$. What is the length of the nonzero vector x? What is the size of F? How many rows and columns of A need to be updated? Note that you need not store all the Householder vectors, v; you may update Q as soon as you computed v. Check your answer using MATLAB `qr` function. Submit your code. (You may implement the standard Householder QR factorization for 3 marks.)

6. (40 marks) Consider an $n \times n$ symmetric matrix generated by `SymMatrix.m` (download from class homepage). The eigenvalues $\{\lambda^{(k)}\}$ are given by:

$$\lambda^{(k)} = 4 \sin^2 \left(\frac{k\pi}{2(n+1)} \right).$$

Note: $\lambda^{(1)} < \lambda^{(2)} < \cdots < \lambda^{(n)}$.

(a) Implement the Householder triangularization algorithm. Create a MATLAB function:

```
[B,Q] = Triangular(A)
```

The input is a symmetric matrix A and the output is a tridiagonal matrix B and transformation matrix Q such that $B = Q^T A Q$. In order to make use of the tridiagonal structure, B should be in sparse format. Suppose B is the Householder transformed A. Then add the following one line of code:

```
B=spdiags(spdiags(B,-1:1),-1:1,n,n);
```

which will create a sparse tridiagonal matrix B. Submit a listing of your code.

(b) Implement the numerical methods: power iteration, Rayleigh quotient iteration, and QR algorithm (no shift). Create the following MATLAB functions:

```
[v,lambda,iter] = PowerIteration(A,v0,maxiter,tol)
[v,lambda,iter] = RayleighQuotient(A,v0,maxiter,tol)
[V,Lambda,iter] = QRIteration(A,maxiter,tol)
```

The first two MATLAB functions take as inputs the matrix A, the initial vector $v0$, the maximum number of iterations $maxiter$ and the tolerance tol, and compute the approximate eigenvector v, approximate eigenvalue $lambda$, and the number of iterations to convergence, $iter$. The third MATLAB function computes all the eigenvectors and eigenvalues of A. The approximate eigenvectors are stored in matrix V and eigenvalues are stored in vector $Lambda$.

For all these methods, first transform the symmetric matrix A to a tridiagonal matrix B using `Triangular` in part (a). Then perform the iterations on B. At the end, remember to transform the eigenvector of B back to the eigenvector of A.

For `RayleighQuotient`, you can use MATLAB backslash \ to solve linear systems. For `QRIteration`, use `BandQR` in Question 5 to compute the QR factorization of A. (You could use MATLAB `qr` for partial credits.) Also, the new matrix $R^{(k)}Q^{(k)}$ may not be exactly tridiagonal due to roundoff errors. For efficiency, you should sparsify the matrix using the MATLAB code in part (a).

For all these methods, the stopping criterion is:

$$\|A\tilde{v} - \tilde{\lambda}\tilde{v}\|_2 < tol,$$

where $\tilde{v}$ and $\tilde{\lambda}$ are the approximate eigenvector and eigenvalue, respectively. For `QRIteration`, the stopping criterion applies to *all* eigenvectors and eigenvalues. Submit all your code.

(c) Compute the eigenvectors and eigenvalues of A using the above methods for $n = 100$. Set $maxiter$ large enough so that the methods converge within the tolerance, $tol = 10^{-8}$. Create a MATLAB program, `EigenMethods.m`, which performs the following:

(i) Use `PowerIteration` to compute the largest eigenvector and eigenvalue of A. The initial vector $v0 = [1, 0, \ldots, 0]^T$.

(ii) Use `RayleighQuotient` to compute an eigenvector and eigenvalue of A. The initial vector $v0 = [1, 0, \ldots, 0]^T$.

(iii) Use `QRIteration` to compute all eigenvectors and eigenvalues of A.

For `PowerIteration`, make a plot of the computed eigenvector. Display the value of the computed eigenvalue and the number of iterations on the title of the plot. Do the same for `RayleighQuotient`. For `QRIteration`, make a plot of all eigenvalues. Also, plot the column vectors v_{20}, v_{40}, v_{60}, v_{80} of V. The title of the plots are the corresponding eigenvalues and number of iterations. Submit `EigenMethods` and the outputs.