

## MATLAB Project # 3 – Due Date: 12/11/2008

You must work each problem in a separate M-file and hand in a printout for each problem. Bring the assignment to room **MTH 4111 at 11AM on Th 12/11**. No late homework will be accepted. Recall that there will be no class on 12/11.

The following three problems describe how eigenvalues can be calculated by *iterative methods* that employ the QR factorization, and provide some theoretical justification. There is no simple way to compute eigenvalues of matrices larger than  $2 \times 2$ . Calculating the roots of the characteristic polynomial does not work well numerically. The most successful algorithm for computing eigenvalues is based on the iterative use of the QR factorization. The MATLAB command `eig` implements this method. Consider the matrices

$$M_1 = \begin{bmatrix} 1 & -2 & 8 \\ 7 & -7 & 6 \\ 5 & 7 & -8 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 4 & -2 & 3 & -7 \\ 1 & 2 & 6 & 8 \\ 8 & 5 & 1 & -5 \\ -5 & 8 & -5 & 3 \end{bmatrix}.$$

**Problem 1** (30 pts). Compute the eigenvalues and eigenvectors of  $M_1$  and  $M_2$  using the command `eig`. Type `[V,D]=eig(Mi)` to perform this task; `help eig` explains the meaning of both V and D.

**Problem 2** (30 pts). Let  $A \in R^{n \times n}$ . Let  $A = Q_0 R_0$  be a QR factorization of  $A$  and create  $A_1 = R_0 Q_0$ , that is change the order of multiplication. Let  $A_1 = Q_1 R_1$  be a QR factorization of  $A_1$  and create  $A_2 = R_1 Q_1$ . Explain why the following statements are true:

- (a)  $A = Q_0 A_1 Q_0^T$ ;
- (b)  $A = (Q_0 Q_1) A_2 (Q_0 Q_1)^T$ ;
- (c)  $Q_0 Q_1$  is orthogonal;
- (d)  $A$ ,  $A_1$ , and  $A_2$  all have the same eigenvalues.

The basic QR algorithm for eigenvalues computes iteratively the QR factorization of the  $m$ -th matrix  $A_m = Q_m R_m$ , and creates the next matrix  $m + 1$  as  $A_{m+1} = R_m Q_m$ , and continues until the entries below the diagonal of  $A_m$  are sufficiently small.

**Problem 3** (40 pts). For both matrices  $M_1$  and  $M_2$ , perform enough steps of the basic QR algorithm to make every entry below the diagonal smaller than 0.1. Record the number of steps for each matrix. Write a `for` or `while` loop to implement the iteration. The basic MATLAB commands for  $M_1$  read as follows:

```
A = M1;
[Q R] = qr(A);
A = R*Q;
```