Exercises for 8.4

Exercise 8.4.1 In each case find the QR-factorization of A.

a. $A = \begin{bmatrix} 1 & -1 \\ -1 & 0 \end{bmatrix}$

b. $A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$

c. $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

d. $A = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & -1 & 0 \end{bmatrix}$

Exercise 8.4.2 Let A and B denote matrices.

a. If A and B have independent columns, show that AB has independent columns. [*Hint*: Theorem 5.4.3.]

b. Show that A has a QR-factorization if and only if A has independent columns.

c. If AB has a QR-factorization, show that the same is true of B but not necessarily A.

[*Hint*: Consider AA^T where $A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$.]

Exercise 8.4.3 If R is upper triangular and invertible, show that there exists a diagonal matrix D with diagonal entries ± 1 such that $R_1 = DR$ is invertible, upper triangular, and has positive diagonal entries.

Exercise 8.4.4 If A has independent columns, let $A = QR$ where Q has orthonormal columns and R is invertible and upper triangular. [Some authors call *this* a QR-factorization of A.] Show that there is a diagonal matrix D with diagonal entries ± 1 such that $A = (QD)(DR)$ is the QR-factorization of A. [*Hint*: Preceding exercise.]

8.5 Computing Eigenvalues

In practice, the problem of finding eigenvalues of a matrix is virtually never solved by finding the roots of the characteristic polynomial. This is difficult for large matrices and iterative methods are much better. Two such methods are described briefly in this section.

The Power Method

In Chapter 3 our initial rationale for diagonalizing matrices was to be able to compute the powers of a square matrix, and the eigenvalues were needed to do this. In this section, we are interested in efficiently computing eigenvalues, and it may come as no surprise that the first method we discuss uses the powers of a matrix.

Recall that an eigenvalue λ of an $n \times n$ matrix A is called a **dominant eigenvalue** if λ has multiplicity 1, and

$$|\lambda| > |\mu| \quad \text{for all eigenvalues } \mu \neq \lambda$$

Any corresponding eigenvector is called a **dominant eigenvector** of A. When such an eigenvalue exists, one technique for finding it is as follows: Let $\mathbf{x}_0$ in $\mathbb{R}^n$ be a first approximation to a dominant eigenvector λ, and compute successive approximations $\mathbf{x}_1, \mathbf{x}_2, \dots$ as follows:

$$\mathbf{x}_1 = A\mathbf{x}_0 \quad \mathbf{x}_2 = A\mathbf{x}_1 \quad \mathbf{x}_3 = A\mathbf{x}_2 \quad \cdots$$

In general, we define

$$\mathbf{x}_{k+1} = A\mathbf{x}_k \quad \text{for each } k \geq 0$$

If the first estimate $\mathbf{x}_0$ is good enough, these vectors $\mathbf{x}_n$ will approximate the dominant eigenvector λ (see below). This technique is called the **power method** (because $\mathbf{x}_k = A^k\mathbf{x}_0$ for each $k \geq 1$). Observe that if $\mathbf{z}$ is any eigenvector corresponding to λ, then

$$\frac{\mathbf{z}\cdot(A\mathbf{z})}{\|\mathbf{z}\|^2} = \frac{\mathbf{z}\cdot(\lambda\mathbf{z})}{\|\mathbf{z}\|^2} = \lambda$$

Because the vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n, \dots$ approximate dominant eigenvectors, this suggests that we define the **Rayleigh quotients** as follows:

$$r_k = \frac{\mathbf{x}_k\cdot\mathbf{x}_{k+1}}{\|\mathbf{x}_k\|^2} \quad \text{for } k \geq 1$$

Then the numbers r_k approximate the dominant eigenvalue λ.

Example 8.5.1

Use the power method to approximate a dominant eigenvector and eigenvalue of $A = \left[\begin{array}{rr} 1 & 1 \\ 2 & 0 \end{array}\right]$.

Solution. The eigenvalues of A are 2 and -1, with eigenvectors $\left[\begin{array}{r} 1 \\ 1 \end{array}\right]$ and $\left[\begin{array}{r} 1 \\ -2 \end{array}\right]$. Take $\mathbf{x}_0 = \left[\begin{array}{r} 1 \\ 0 \end{array}\right]$ as the first approximation and compute $\mathbf{x}_1, \mathbf{x}_2, \dots,$ successively, from $\mathbf{x}_1 = A\mathbf{x}_0, \mathbf{x}_2 = A\mathbf{x}_1, \dots$. The result is

$$\mathbf{x}_1 = \left[\begin{array}{r} 1 \\ 2 \end{array}\right], \ \mathbf{x}_2 = \left[\begin{array}{r} 3 \\ 2 \end{array}\right], \ \mathbf{x}_3 = \left[\begin{array}{r} 5 \\ 6 \end{array}\right], \ \mathbf{x}_4 = \left[\begin{array}{r} 11 \\ 10 \end{array}\right], \ \mathbf{x}_3 = \left[\begin{array}{r} 21 \\ 22 \end{array}\right], \ \dots$$

These vectors are approaching scalar multiples of the dominant eigenvector $\left[\begin{array}{r} 1 \\ 1 \end{array}\right]$. Moreover, the Rayleigh quotients are

$$r_1 = \tfrac{7}{5}, \ r_2 = \tfrac{27}{13}, \ r_3 = \tfrac{115}{61}, \ r_4 = \tfrac{451}{221}, \ \dots$$

and these are approaching the dominant eigenvalue 2.

To see why the power method works, let $\lambda_1, \lambda_2, \dots, \lambda_m$ be eigenvalues of A with λ_1 dominant and let $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m$ be corresponding eigenvectors. What is required is that the first approximation $\mathbf{x}_0$ be a linear combination of these eigenvectors:

$$\mathbf{x}_0 = a_1\mathbf{y}_1 + a_2\mathbf{y}_2 + \cdots + a_m\mathbf{y}_m \quad \text{with } a_1 \neq 0$$

If $k \geq 1$, the fact that $\mathbf{x}_k = A^k\mathbf{x}_0$ and $A^k\mathbf{y}_i = \lambda_i^k\mathbf{y}_i$ for each i gives

$$\mathbf{x}_k = a_1\lambda_1^k\mathbf{y}_1 + a_2\lambda_2^k\mathbf{y}_2 + \cdots + a_m\lambda_m^k\mathbf{y}_m \quad \text{for } k \geq 1$$

Hence

$$\frac{1}{\lambda_1^k}\mathbf{x}_k = a_1\mathbf{y}_1 + a_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\mathbf{y}_2 + \cdots + a_m\left(\frac{\lambda_m}{\lambda_1}\right)^k\mathbf{y}_m$$

The right side approaches $a_1\mathbf{y}_1$ as k increases because λ_1 is dominant $\left(\left|\frac{\lambda_i}{\lambda_1}\right| < 1 \text{ for each } i > 1\right)$. Because $a_1 \neq 0$, this means that $\mathbf{x}_k$ approximates the dominant eigenvector $a_1\lambda_1^k\mathbf{y}_1$.

The power method requires that the first approximation $\mathbf{x}_0$ be a linear combination of eigenvectors. (In Example 8.5.1 the eigenvectors form a basis of $\mathbb{R}^2$.) But even in this case the method fails if $a_1 = 0$, where a_1 is the coefficient of the dominant eigenvector (try $\mathbf{x}_0 = \left[\begin{array}{r} -1 \\ 2 \end{array}\right]$ in Example 8.5.1). In general, the rate of convergence is quite slow if any of the ratios $\left|\frac{\lambda_i}{\lambda_1}\right|$ is near 1. Also, because the method requires repeated multiplications by A, it is not recommended unless these multiplications are easy to carry out (for example, if most of the entries of A are zero).

QR-Algorithm

A much better method for approximating the eigenvalues of an invertible matrix A depends on the factorization (using the Gram-Schmidt algorithm) of A in the form

$$A = QR$$

where Q is orthogonal and R is invertible and upper triangular (see Theorem 8.4.2). The **QR-algorithm** uses this repeatedly to create a sequence of matrices $A_1 = A,\ A_2,\ A_3,\ \dots,$ as follows:

1. Define $A_1 = A$ and factor it as $A_1 = Q_1R_1$.
2. Define $A_2 = R_1Q_1$ and factor it as $A_2 = Q_2R_2$.
3. Define $A_3 = R_2Q_2$ and factor it as $A_3 = Q_3R_3$.

$$\vdots$$

In general, A_k is factored as $A_k = Q_kR_k$ and we define $A_{k+1} = R_kQ_k$. Then A_{k+1} is similar to A_k [in fact, $A_{k+1} = R_kQ_k = (Q_k^{-1}A_k)Q_k$], and hence each A_k has the same eigenvalues as A. If the eigenvalues of A are real and have distinct absolute values, the remarkable thing is that the sequence of matrices $A_1, A_2, A_3, \dots$ converges to an upper triangular matrix with these eigenvalues on the main diagonal. [See below for the case of complex eigenvalues.]

Example 8.5.2

If $A = \left[\begin{array}{rr} 1 & 1 \\ 2 & 0 \end{array}\right]$ as in Example 8.5.1, use the QR-algorithm to approximate the eigenvalues.

Solution. The matrices A_1, A_2, and A_3 are as follows:

$$A_1 = \left[\begin{array}{rr} 1 & 1 \\ 2 & 0 \end{array}\right] = Q_1R_1 \quad \text{where } Q_1 = \frac{1}{\sqrt{5}}\left[\begin{array}{rr} 1 & 2 \\ 2 & -1 \end{array}\right] \text{ and } R_1 = \frac{1}{\sqrt{5}}\left[\begin{array}{rr} 5 & 1 \\ 0 & 2 \end{array}\right]$$

$$A_2 = \frac{1}{5}\left[\begin{array}{rr} 7 & 9 \\ 4 & -2 \end{array}\right] = \left[\begin{array}{rr} 1.4 & -1.8 \\ -0.8 & -0.4 \end{array}\right] = Q_2R_2$$

where $Q_2 = \frac{1}{\sqrt{65}}\left[\begin{array}{rr} 7 & 4 \\ 4 & -7 \end{array}\right]$ and $R_2 = \frac{1}{\sqrt{65}}\left[\begin{array}{rr} 13 & 11 \\ 0 & 10 \end{array}\right]$

$$A_3 = \frac{1}{13}\left[\begin{array}{rr} 27 & -5 \\ 8 & -14 \end{array}\right] = \left[\begin{array}{rr} 2.08 & -0.38 \\ 0.62 & -1.08 \end{array}\right]$$

This is converging to $\left[\begin{array}{rr} 2 & * \\ 0 & -1 \end{array}\right]$ and so is approximating the eigenvalues 2 and -1 on the main diagonal.

It is beyond the scope of this book to pursue a detailed discussion of these methods. The reader is referred to J. M. Wilkinson, *The Algebraic Eigenvalue Problem* (Oxford, England: Oxford University Press, 1965) or G. W. Stewart, *Introduction to Matrix Computations* (New York: Academic Press, 1973). We conclude with some remarks on the QR-algorithm.

Shifting. Convergence is accelerated if, at stage k of the algorithm, a number s_k is chosen and $A_k - s_kI$ is factored in the form Q_kR_k rather than A_k itself. Then

$$Q_k^{-1}A_kQ_k = Q_k^{-1}(Q_kR_k + s_kI)Q_k = R_kQ_k + s_kI$$

so we take $A_{k+1} = R_kQ_k + s_kI$. If the shifts s_k are carefully chosen, convergence can be greatly improved.

Preliminary Preparation. A matrix such as

$$\left[\begin{array}{ccccc} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{array}\right]$$

is said to be in **upper Hessenberg** form, and the QR-factorizations of such matrices are greatly simplified. Given an $n \times n$ matrix A, a series of orthogonal matrices $H_1, H_2, \dots, H_m$ (called **Householder matrices**) can be easily constructed such that

$$B = H_m^T \cdots H_1^T A H_1 \cdots H_m$$

is in upper Hessenberg form. Then the QR-algorithm can be efficiently applied to B and, because B is similar to A, it produces the eigenvalues of A.

Complex Eigenvalues. If some of the eigenvalues of a real matrix A are not real, the QR-algorithm converges to a block upper triangular matrix where the diagonal blocks are either 1×1 (the real eigenvalues) or 2×2 (each providing a pair of conjugate complex eigenvalues of A).

Exercises for 8.5

Exercise 8.5.1 In each case, find the exact eigenvalues and determine corresponding eigenvectors. Then start with $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and compute $\mathbf{x}_4$ and r_3 using the power method.

a. $A = \begin{bmatrix} 2 & -4 \\ -3 & 3 \end{bmatrix}$

b. $A = \begin{bmatrix} 5 & 2 \\ -3 & -2 \end{bmatrix}$

c. $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

d. $A = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix}$

Exercise 8.5.2 In each case, find the exact eigenvalues and then approximate them using the QR-algorithm.

a. $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$

b. $A = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix}$

Exercise 8.5.3 Apply the power method to $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, starting at $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Does it converge? Explain.

Exercise 8.5.4 If A is symmetric, show that each matrix A_k in the QR-algorithm is also symmetric. Deduce that they converge to a diagonal matrix.

Exercise 8.5.5 Apply the QR-algorithm to $A = \begin{bmatrix} 2 & -3 \\ 1 & -2 \end{bmatrix}$. Explain.

Exercise 8.5.6 Given a matrix A, let A_k, Q_k, and R_k, $k \geq 1$, be the matrices constructed in the QR-algorithm. Show that $A_k = (Q_1 Q_2 \cdots Q_k)(R_k \cdots R_2 R_1)$ for each $k \geq 1$ and hence that this is a QR-factorization of A_k.
[*Hint*: Show that $Q_k R_k = R_{k-1} Q_{k-1}$ for each $k \geq 2$, and use this equality to compute $(Q_1 Q_2 \cdots Q_k)(R_k \cdots R_2 R_1)$ "from the centre out." Use the fact that $(AB)^{n+1} = A(BA)^n B$ for any square matrices A and B.]

8.6 The Singular Value Decomposition

When working with a square matrix A it is clearly useful to be able to "diagonalize" A, that is to find a factorization $A = Q^{-1}DQ$ where Q is invertible and D is diagonal. Unfortunately such a factorization may not exist for A. However, even if A is not square gaussian elimination provides a factorization of the form $A = PDQ$ where P and Q are invertible and D is diagonal—the Smith Normal form (Theorem 2.5.3). However, if A is real we can choose P and Q to be *orthogonal* real matrices and D to be real. Such a factorization is called a **singular value decomposition (SVD)** for A, one of the most useful tools in applied linear algebra. In this Section we show how to explicitly compute an SVD for any real matrix A, and illustrate some of its many applications.

We need a fact about two subspaces associated with an $m \times n$ matrix A:

$$\text{im } A = \{A\mathbf{x} \mid \mathbf{x} \text{ in } \mathbb{R}^n\} \qquad \text{and} \qquad \text{col } A = \text{span}\,\{\mathbf{a} \mid \mathbf{a} \text{ is a column of } A\}$$

Then im A is called the **image** of A (so named because of the linear transformation $\mathbb{R}^n \to \mathbb{R}^m$ with $\mathbf{x} \mapsto A\mathbf{x}$); and col A is called the **column space** of A (Definition 5.10). Surprisingly, these spaces are equal:

Lemma 8.6.1

For any $m \times n$ matrix A, $\text{im } A = \text{col } A$.