Chapter 4 of Calculus^{++}: The Non–symmetric Eigenvalue Problem

by

Eric A Carlen
Professor of Mathematics
Georgia Tech

Table of Contents

Overview

This chapter concerns the *non symmetric eigenvalue problem.* Here we shall develop a means for computing the eigenvalues of an arbitrary square matrix. This problem is fundamentally important in the calculus of several variables since many applications require the computation of the eigenvalues of the Jacobian of a function $\mathbf{F}$ from $I\!R^n$ to $I\!R^n$. Unlike Hessians, Jacobian matrices need not be symmetric.

In the last chapter, we saw how to find the eigenvalues of a symmetric matrix by a progressive diagonalization procedure. This cannot work in the general non–symmetric case since non–symmetric matrices may not be diagonalizable.

However, as we shall see, something almost as good is true for every square matrix A: Every square matrix A is similar to an upper triangular matrix T. It is not hard to see that the eigenvalues of any upper triangular matrix T are simply the diagonal entries of T. Since A is similar to T, it has the same eigenvalues. Hence once we have found T, we have found the eigenvalues of A.

There is another complication to deal with though. While the eigenvalues of a symmetric matrix are always real, this need not be the case for a non–symmetric matrix. Hence we shall be forced to work with complex numbers in this chapter. If all of the eigenvalues happen to be real, then we shall see that not only is A similar to an upper triangular matrix T; we have

$$A = QTQ^{-1}$$

where Q is an *orthogonal matrix.* Such a factorization of A is known as a *Schur factorization.*

The fact that Q is very important in numerical calculation. Here is why: *Entries of orthogonal matrices are never, ever large.* In fact, an $n \times n$ matrix Q is orthogonal if and only is its column are an orthonormal basis for $I\!R^n$. In particular, the columns are unit vectors, and so no entry can have an absolute value any greater than 1. Also, Q^{-1} is simply Q^t, another orthogonal matrix. So it's entries cannot be large either. As we shall explain, this means that the factorization $A = QTQ^{-1}$ can be used safely in numerical computations.

When complex eigenvalues show up, we must use *unitary matrices.* These are the complex analog of orthogonal matrices; they describe isometries in the space of n dimensional complex vectors C^n. The transition from real to complex vectors is not difficult, and we make it in the second section.

The next three sections are devoted to developing a means to compute a Schur factorization $A = QTQ^{-1}$. This involves a beautiful iterative application of the QR factorization that we explain form the beginning. The final section is devoted to the use of Schur factorizations in computing matrix exponentials. We shall see that Schur factorization provides a practical method for computing the exponential of any square matrix.

Section 1. Schur factorization

1.1 The non–symmetric eigenvalue problem

We now know how to find the eigenvalues and eigenvectors of *any* symmetric $n \times n$ matrix, no matter how large. This is useful in the the calculus of several variables since Hessian matrices are always symmetric. Hence we have the means to find the eigenvectors and eigenvalues of the Hessian of a function of any number of variables. We can do this by Jacobi's progressive diagonalization method. We have effectively solved what is known as *the symmetric eigenvalue problem.*

The other kind of matrices that frequently arises in the calculus of several variables are Jacobian matrices. These need not be symmetric, and so we cannot use the methods of the last chapter to find their eigenvalues and eigenvectors. As we shall see later in the course, many problems in the description and prediction of motion require us to determine the eigenvalues of Jacobian matrices. What do we do with them?

Our approach will have to be different. We cannot hope to use any sort of progressive diagonalization method since not every non symmetric matrix can be diagonalized. However, there is something we can do that is almost as good: *We can upper triangularize any square matrix.*

1.2 What is the Schur factorization?

It turns out that every $n \times n$ matrix A can be factored as

$$A = Q^{-1}TQ \tag{1.1}$$

where Q is unitary, and where T is upper triangular. Unitary matrices are the complex analogs of orthogonal matrices, and in case all of the eigenvalues of A happen to be real, Q will be an orthogonal matrix. To say that T is upper triangular just means that $T_{i,j} = 0$ for $i > j$. That is, every entry below the diagonal is zero.

As far as finding the eigenvalues of A is concerned, the point is that:

• *If $A = QTQ^{-1}$ where T is upper triangular, then the eigenvalues of A are exactly the diagonal entries of T*

To see this, recall that similar matrices have the same eigenvalues, and (1.1)says that A and T are similar. Also $T - tI$ is upper triangular, so its determinant is the product of its diagonal entries – $(T_{1,1} - t)(T_{2,2} - t)\cdots(T_{n,n} - t)$. The roots of the characteristic polynomial are plainly the diagonal entries of T.

The fact that *every* square matrix A can be factored as $A = QTQ^{-1}$ was discovered by Schur, and hence such a factorization is called a *Schur factorization.* Schur factorization is the substitute for diagonalization, which is not always possible. For instance, the matrix $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ cannot be diagonalized. The only eigenvectors of this matrix are multiples of $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and so there cannot be a basis of eigenvalues.

As we shall see, Schur factorization is a very good substitute for diagonalization. For instance, using a Schur factorization it is easy to compute e^{tA}, which will be important when it comes to dealing with problems in the description and prediction of motion.

1.3 The 2×2 case

Without further ado, let's compute a Schur factorization. Let $A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$. As you can see the entries in each row add up to 5, and this means that with $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$,

$$A\mathbf{v} = 5\mathbf{v} \ .$$

Finding one eigenvector – one way or another – is the key to computing a Schur factorization.

We now build an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$ whose first element is an eigenvector of A. To do this, normalize $\mathbf{v}$: Let

$$\mathbf{u}_1 = \frac{1}{|\mathbf{v}|}\mathbf{v} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

and let

$$\mathbf{u}_2 = \mathbf{u}_1^{\perp} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Let Q be the orthogonal matrix $Q = [\mathbf{u}_1, \mathbf{u}_2]$. We have

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \ .$$

Then,

$$AQ = A[\mathbf{u}_1, \mathbf{u}_2] = [A\mathbf{u}_1, A\mathbf{u}_2] = [5\mathbf{u}_1, A\mathbf{u}_2] \ .$$

In the same way,

$$Q^{-1}AQ = Q^{-1}[5\mathbf{u}_1, A\mathbf{u}_2] = [5Q^{-1}\mathbf{u}_1, Q^{-1}A\mathbf{u}_2] \ .$$

Since Q is orthogonal, $Q^{-1} = Q^t$, and

$$Q^{-1}\mathbf{u}_1 = \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{u}_1 \\ \mathbf{u}_2 \cdot \mathbf{u}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ .$$

Hence,

$$Q^{-1}AQ = \begin{bmatrix} 5 & * \\ 0 & * \end{bmatrix} \tag{1.2}$$

where we have not yet computed the second column. *But whatever it is, the result is an upper triangular matrix.*

Multiplying everything out, we find

$$Q^{-1}AQ = \begin{bmatrix} 5 & 2 \\ 0 & 1 \end{bmatrix}$$

and so with $T = \begin{bmatrix} 5 & 2 \\ 0 & 1 \end{bmatrix}$, $A = QTQ^{-1}$.

This gives us our Schur factorization of A. In particular, we see that the second eigenvalue of A is 1. Of course, we know how to find the eigenvalues of 2×2 matrices, so this by itself is no so exciting. But as we shall see, the Schur factorization makes it easy to answer all of the questions about A that arise if A is the Jacobian matrix of some non linear transformation. So computing Schur factorizations does turn out useful, even in the 2×2 case.

If you look back over the calculation, you see that what made it work was that the first column of the orthogonal matrix Q was an eigenvector of A. That is all we really used to arrive at (1.2); the fact that the eigenvalue happened to be 5 was inconsequential. This example leads us to the following result:

Theorem *Let A be a 2×2 matrix, and suppose that A has a real eigenvalue μ. Let $\mathbf{u}_1$ be a unit vector with $A\mathbf{u}_1 = \mu_1\mathbf{u}_1$, and let $\mathbf{u}_2 = \mathbf{u}_1^{\perp}$. Then with $Q = [\mathbf{u}_1, \mathbf{u}_2]$, Q is orthogonal, and $Q^{-1}AQ = T$ is upper triangular.*

Example 1 (Computation of a Schur Factorization) Let $A = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}$. To compute a Schur factorization, we need to find one eigenvector. Computing the characteristic polynomial, we find it is

$$t^2 - 4t - 5 = (t-5)(t+1) \ .$$

Hence the eigenvalues are 5 and -1. To find an eigenvector with the eigenvalue 5, we form

$$A - 5I = \begin{bmatrix} -2 & 2 \\ 4 & -4 \end{bmatrix} \ .$$

Clearly $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is in the kernel of this matrix, and so

$$\mathbf{u}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

is a normalized eigenvector with eigenvalue 5. Taking $\mathbf{u}_2 = \mathbf{u}_1^{\perp}$, we have

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \ .$$

Multiplying out $Q^{-1}AQ$, remembering that $Q^{-1} = Q^t$, we find

$$Q^{-1}AQ = \begin{bmatrix} 5 & -2 \\ 0 & -1 \end{bmatrix} \ .$$

Hence the upper triangular matrix T is $T = \begin{bmatrix} 5 & -2 \\ 0 & -1 \end{bmatrix}$. As you see, the second diagonal entry is -1, which it must be according to the theory developed above, since the second eigenvalue is -1.

Example 2 (Alternate computation of a Schur Factorization) Once again let $A = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}$ as in the first example. There we found the eigenvalues to be 5 and -1. This time, let's work with an eigenvector for the eigenvalue -1.

To find an eigenvector with the eigenvalue -1, we form

$$A - 1I = \begin{bmatrix} 4 & 2 \\ 4 & 2 \end{bmatrix} .$$

Clearly $\mathbf{v} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ is in the kernel of this matrix, and so

$$\mathbf{u}_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

is a normalized eigenvector with eigenvalue -1. Taking $\mathbf{u}_2 = \mathbf{u}_1^\perp$, we have

$$Q = \frac{1}{\sqrt{5}} \begin{bmatrix} -1 & -2 \\ 2 & -1 \end{bmatrix} .$$

Multiplying out $Q^{-1}AQ$, remembering that $Q^{-1} = Q^t$, we find

$$Q^{-1}AQ = \begin{bmatrix} -1 & -2 \\ 0 & -5 \end{bmatrix} .$$

Hence the upper triangular matrix T is $T = \begin{bmatrix} -1 & -2 \\ 0 & -5 \end{bmatrix}$. As you see, the second diagonal entry is 5, which it must be according to the theory developed above, since the second eigenvalue is 5.

As you see, as long as you can find one eigenvector – any one will do – you can find the Schur factorization. Now every square matrix has at least one eigenvalue, and hence there is always at least one eigenvector to be found. The problem is that even if the matrix has real entries, the eigenvalue may be complex, and then the corresponding eigenvector will be complex. To have a general method, we must deal with complex eigenvalues and eigenvectors. This does not complicate matters very much, as we explain in the next section.

Problems

1 Let $A = \begin{bmatrix} -3 & 4 \\ -6 & 7 \end{bmatrix}$. Compute a Schur factorization of A.

2 Let $A = \begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix}$. Compute a Schur factorization of A.

3 Let $A = \begin{bmatrix} -3 & 5 \\ -2 & 4 \end{bmatrix}$. Compute a Schur factorization of A.

4 Let $A = \begin{bmatrix} 1 & 2 \\ -3 & 5 \end{bmatrix}$. Compute a Schur factorization of A.

Section 1. Complex eigenvectors and the geometry of C^n

1.1 Why get complicated?

Our goal in this chapter is to explain how one can effectively compute the eigenvalues of non symmetric matrices. In general, the eigenvectors of a non symmetric matrix will be complex, and therefore, the eigenvectors themselves will in general have complex entries. To work effectively with complex eigenvalues and eigenvectors we need to extend some of our geometric and algebraic understanding of real vectors to the complex case.

By the fundamental theorem of algebra, every polynomial has roots in the complex plane. Therefore, as long as we allow complex eigenvalues and complex eigenvectors, every matrix A has at least one eigenvalue.

Example 1 (Complex eigenvalues) Let $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. The characteristic polynomial $p_A(t)$ is

$$p_A(t) = \det\left(\begin{bmatrix} -t & 1 \\ -1 & -t \end{bmatrix}\right) = t^2 + 1 \ .$$

Evidently, there are no real solutions to $p_A(t) = 0$. However, we do have the two complex solutions, $\mu_1 = i$ and $\mu_2 = -i$, where i denotes $\sqrt{-1}$. These are the eigenvalues of A, and to find the eigenvectors with eigenvalue i, we form

$$A - iI = \begin{bmatrix} -i & 1 \\ -1 & -i \end{bmatrix} \ .$$

The kernel of this matrix is spanned by $\mathbf{v}_1 = \begin{bmatrix} -i \\ 1 \end{bmatrix}$, and hence this complex vector satisfies $A\mathbf{v}_1 = \mu_1\mathbf{v}_1$. Likewise,

$$A + iI = \begin{bmatrix} i & 1 \\ -1 & i \end{bmatrix} \ ,$$

and so the kernel of this matrix is spanned by $\mathbf{v}_2 = \begin{bmatrix} i \\ 1 \end{bmatrix}$, and hence this complex vector satisfies $A\mathbf{v}_2 = \mu_2\mathbf{v}_2$. So as long as we are willing to admit complex eigenvalues and eigenvectors with complex entries, then every $n \times n$ matrix has at least one eigenvalue.

As we have seen, even if all of the entries of an $n \times n$ matrix A are real numbers, all of the eigenvalues may be complex, or even purely imaginary. This means that the corresponding eigenvectors will have complex entries.

Definition (C^n, complex vectors) The set of all vectors $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$ where $z_1, z_2, \dots, z_n$ are complex numbers is denoted C^n, and called *complex n dimensional space.*

1.2 Algebra and geometry in C^n

The algebra of vectors in C^n is just the same as in $I\!R^n$, with the one exception that now we allow the entries to be complex numbers and can multiply vectors by complex numbers. We still add the vectors entry by entry, etc., as before.

While we are at it, we may as well consider $m \times n$ matrices with complex entries. Again, all of the algebra is just the same as before – all of our formulas generalize to to the case in which the entries are complex numbers, and the proofs are the same, since addition and multiplication of complex numbers has the same associative, commutative, and distributive properties as does the addition and multiplication of real numbers.

For the exact same reason, all of the theorems we have proved about solving linear systems of equation by row reduction still hold – with the same proofs – if we allow the coefficients and variables to take on complex values as well.

The geometry of C^n is another matter. We cannot define the length of a vector $\mathbf{z}$ in C^n as the positive square root of $\mathbf{z} \cdot \mathbf{z}$, since in general, $\mathbf{z} \cdot \mathbf{z}$ will be a complex number, and there won't be any positive square root.

When $z = x + iy$ is a single complex number, the *modulus* of z, also called the *length* of z, is denoted by $|z|$ and is defined by

$$|z|^2 = x^2 + y^2 \ .$$

If, as usual, we identify the set of complex numbers with the plane $I\!R^2$, so that $z = x + iy$ is identified with the vector $\begin{bmatrix} x \\ y \end{bmatrix}$, then $|z|$ coincides with the length of the vector $\begin{bmatrix} x \\ y \end{bmatrix}$.

Recall that the complex conjugate $\bar{z}$ of $z = x + iy$ is defined by $\bar{z} = x - iy$, and

$$\bar{z}z = x^2 + y^2 = |z|^2 \ .$$

We extend the complex conjugate to C^n as follows: If $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$, then $\bar{\mathbf{z}}$ is the complex vector

$$\bar{\mathbf{z}} = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \\ \vdots \\ \bar{z}_n \end{bmatrix} \ .$$

With this definition,

$$\begin{aligned} \bar{\mathbf{z}} \cdot \mathbf{z} &= \bar{z}_1 z_1 + \bar{z}_2 z_2 + \cdots + \bar{z}_n z_n \\ &= (x_1^2 + y_1^2) + (x_2^2 + y_2^2) + \cdots (x_n^2 + y_n^2) \ . \end{aligned}$$

This *is* a positive number, and we can therefore define

$$|\mathbf{z}| = \sqrt{\bar{\mathbf{z}} \cdot \mathbf{z}} \ .$$

We can then define the *distance* between two vectors $\mathbf{w}$ and $\mathbf{z}$ to be the length of their difference, $|\mathbf{w} - \mathbf{z}|$ as before. Of course, if we are going to refer to this as "distance", we had better be able to prove that it has the properties of distance function, as explained in Chapter One. In particular, we had better be able to prove that it satisfies the triangle inequality. This is the case, and as with $I\!R^n$, the key is the Schwarz inequality. First, we make a definition.

Definition (Inner product in C^n) Given two vectors $\mathbf{w}$ and $\mathbf{z}$ in C^n, we define their *inner product*, denoted $\langle \mathbf{w}, \mathbf{z} \rangle$, by

$$\langle \mathbf{w}, \mathbf{z} \rangle = \bar{\mathbf{w}} \cdot \mathbf{z} \ .$$

Notice that $\langle \mathbf{z}, \mathbf{w} \rangle$ is the complex conjugate of $\langle \mathbf{w}, \mathbf{z} \rangle$; order matters in inner products! The Schwarz inequality then is

$$|\langle \mathbf{w}, \mathbf{z} \rangle \leq |\mathbf{w}||\mathbf{z}| \ .$$

It is left as an exercise to adapt the proofs of the Schwarz inequality, and then the triangle inequality form $I\!R^n$ to C^n.

One final definition is required to extend all of our theorems to the complex case. Recall the key property of the transpose:

$$\mathbf{x} \cdot A\mathbf{y} = (A^t\mathbf{x}) \cdot \mathbf{y} \ .$$

This is still true for complex vectors and matrices. But it no longer has geometric meaning in the complex case, since there it is the inner product, and not the dot product that has geometric meaning. Here is the geometric analog of the transpose in the complex case:

Definition (Adjoint of a complex matrix) Given an $m \times n$ matrix A with complex entries, its *adjoint* A^* is defined by

$$A^*_{i,j} = \bar{A_{j,i}} \ .$$

A matrix A that saistifies $A^* = A$ is called *self afjoint.*

That is, the entries of A^* are the complex conjugates of the entries of A^t, the transpose of A.

To see why this is the "right" definition,

$$\begin{aligned} \langle \mathbf{w}, A\mathbf{z} \rangle &= \bar{\mathbf{w}} \cdot A\mathbf{z} \\ &= A^t\bar{\mathbf{w}} \cdot \mathbf{z} \\ &= \bar{A^*\mathbf{w}} \cdot \mathbf{z} \\ &= \langle A^*\mathbf{w}, \mathbf{z} \rangle \ . \end{aligned}$$

The relation

$$\langle \mathbf{w}, A\mathbf{z} \rangle = \langle A^*\mathbf{w}, \mathbf{z} \rangle$$

is the geometric analog of the fundamental property of the transpose.

With the changes we have described here, every theorem we have proved for real vectors and matrices extends to complex vectors and matrices. In particular, any collection of orthonormal vectors in C^n is linearly independent.

1.3 Unitary matrices

Recall that an $n \times n$ matrix U with real entries is orthogonal in case $U^t U = I$. Since the entries of U are real, $U^t = U^*$, and so $U^* U = I$. Complex matrices with the latter property are called *unitary.*

Definition (Unitary matrix) An $m \times n$ matrix U with complex entries is *unitary* in case $U^* U = I$.

Orthogonal matrices preserve the length of vectors in $I\!R^n$; unitary matrices preserve the length of vectors in C^n. Indeed, for any $\mathbf{z}$ in C^n, we have

$$|U\mathbf{z}|^2 = \langle U\mathbf{z}, U\mathbf{z}\rangle = \langle U^*U\mathbf{z}, \mathbf{z}\rangle = \langle \mathbf{z}, \mathbf{z}\rangle = |\mathbf{z}|^2 \ .$$

It is left as an exercise to show that for any matrix A, the determinant of A^* is the complex conjugate of the determinant of A. That is,

$$\det(A^*) = (\det(A))^* \ .$$

Since $I = U^*U$,

$$1 = (\det(U))^*(\det(U)) = |\det(U)|^2 \ .$$

This means that $\det(U)$ is a complex number of modulus one. This is one justification for calling such matrices unitary, But there is more: Suppose that U is unitary and $\mathbf{z}$ is any eigenvector of U with eigenvalue μ. Then

$$|\mu||\mathbf{z}| = |\mu\mathbf{z}| = |U\mathbf{z}| = |\mathbf{z}| \ .$$

Hence we see that

$$|\mu| = 1 \ .$$

In other words, if U is unitary, then all of the eigenvalues of U are complex numbers of unit modulus. However, not every matrix whose eigenvalues are all of unit modulus is unitary; the fundaments property is that unitary matrices preserve the length of vectors in C^n.

Problems

1.1 Compute $|\mathbf{z}|$ for $\mathbf{z} = \begin{bmatrix} 1+i \\ -3+2i \\ 1-4i \end{bmatrix}$.

1.2 Compute $|\mathbf{z}|$ for $\mathbf{z} = \begin{bmatrix} 5-i \\ i \\ 5 \\ 1-3i \end{bmatrix}$.

1.3 Show that if $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ is any vector in C^2, then

$$\mathbf{z}^\perp = \begin{bmatrix} -z_2^* \\ z_1^* \end{bmatrix}$$

is such that

$$|\mathbf{z}^\perp| = |\mathbf{z}| \qquad \text{and} \qquad \langle \mathbf{z}, \mathbf{z}^\perp\rangle = 0 \ .$$

Hence, given any unit vector $\mathbf{z}$ in C^2, $\{\mathbf{z}, \mathbf{z}^\perp\}$ is an orthonormal basis of C^2.

1.4 Let $\mathbf{z} = \begin{bmatrix} 1+2i \\ i \\ 3-i \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} 3+4i \\ 4+3i \\ 2+11i \end{bmatrix}$. Compute $\langle \mathbf{w}, \mathbf{z} \rangle$.

1.5 Let $\mathbf{z} = \begin{bmatrix} 1+2i \\ i \\ 3-i \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} 3+4i \\ 4+3i \\ 2+11i \end{bmatrix}$. Compute $\langle \mathbf{w}, \mathbf{z} \rangle$.

1.6 Find a vector $\mathbf{w}$ in C^2 that is orthogonal to $\mathbf{u} = \begin{bmatrix} 2+4i \\ 4i \end{bmatrix}$. Then find a 2×2 unitary matrix whose first column is a real multiple of $\mathbf{u}$.

1.7 Find a vector $\mathbf{w}$ in C^2 that is orthogonal to $\mathbf{u} = \begin{bmatrix} 7+6i \\ 6 \end{bmatrix}$. Then find a 2×2 unitary matrix whose first column is a real multiple of $\mathbf{u}$.

1.8 Compute the adjoint of $A = \begin{bmatrix} 1+i & i \\ 2-i & 3+i \end{bmatrix}$.

1.9 Compute the adjoint of $A = \begin{bmatrix} 3+2i & 5+2i \\ 1-7i & 2-i \end{bmatrix}$.

Section 3: Householder Reflections

3.1 Reflection matrices

The main problem we wish to solve in this section is to find a good way to write down a unitary matrix whose first column is any given vector. There is a very nice geometric solution to this problem: We can use the matrix that reflects the given vector onto $\mathbf{e}_1$. To explain why this is the case, and how to write down this matrix, consider the problem in $I\!R^2$ first so that we can draw pictures.

At first, we will consider a problem that is slightly more restricted, and also slightly more general: We will consider only real vectors and orthogonal matrices, but we will not require the second vector to be $\mathbf{e}_1$. We ask:

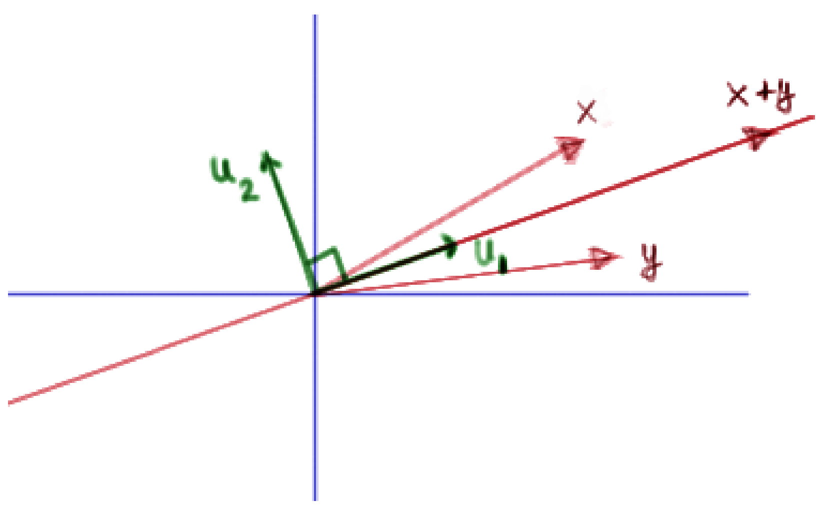
• *Given two vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *in* $I\!R^2$ *of the same length, but not on the same line, what matrix* M *corresponds to the reflection in* $I\!R^2$ *that carries* $\mathbf{x}$ *onto* $\mathbf{y}$*?*

Reflections are isometric transformations, so it is not possible to reflect $\mathbf{x}$ onto $\mathbf{y}$ unless both vectors have the same length. As long as the vectors have the same length, we can easily find the reflection.

The line about which we should reflect is the line through the origin and $\mathbf{y}+\mathbf{x}$. The line orthogonal to this one is the line through the origin and $\mathbf{y}-\mathbf{x}$. Indeed,

$$(\mathbf{y}-\mathbf{x})\cdot(\mathbf{y}+\mathbf{x}) = |\mathbf{y}|^2 - |\mathbf{x}|^2 = 0\ . \tag{3.1}$$

Hence we define $\mathbf{u}_1 = \dfrac{1}{|\mathbf{y}+\mathbf{x}|}(\mathbf{y}+\mathbf{x})$ and $\mathbf{u}_2 = \dfrac{1}{|\mathbf{y}-\mathbf{x}|}(\mathbf{y}-\mathbf{x})$, and this makes $\{\mathbf{u}_1,\mathbf{u}_2\}$ an orthonormal basis.



Let $\mathbf{w}$ be any vector in $I\!R_2$, and it suppose that it has coordinates a_1 and a_2 with respect to the basis $\{\mathbf{u}_1,\mathbf{u}_2\}$. That is,

$$\mathbf{w} = a_1\mathbf{u}_1 + a_2\mathbf{u}_2\ .$$

Since $\{\mathbf{u}_1, \mathbf{u}_2\}$ is an orthonormal basis, the coordinates a_1 and a_2 are given by

$$a_1 = \mathbf{u}_1 \cdot \mathbf{w} \qquad \text{and} \qquad a_2 = \mathbf{u}_2 \cdot \mathbf{w} \ .$$

Also since M should leave $\mathbf{u}_1$ fixed, and should change the sign of $\mathbf{u}_2$, we have

$$\begin{aligned} M\mathbf{w} &= M(a_1\mathbf{u}_1 + a_2\mathbf{u}_2) \\ &= a_1 M\mathbf{u}_1 + a_2 M\mathbf{u}_2 \\ &= a_1\mathbf{u}_1 - a_2\mathbf{u}_2 \\ &= (a_1\mathbf{u}_1 + a_2\mathbf{u}_2) - 2a_2\mathbf{u}_2 \\ &= \mathbf{w} - 2(\mathbf{u}_2 \cdot \mathbf{w})\mathbf{u}_2 \ . \end{aligned} \tag{3.2}$$

To write this in a convenient form, regard $\mathbf{u}_2$ as a 2×1 matrix. Then $\mathbf{u}_1^t$ is a 1×2 matrix, and we can form the matrix product $\mathbf{u}_2^t\mathbf{w}$, which gives a 1×1 matrix; i.e., a number. In fact, as you can easily check,

$$\mathbf{u}_2^t\mathbf{w} = \mathbf{u}_2 \cdot \mathbf{w} \ .$$

Therefore,

$$(\mathbf{u}_2 \cdot \mathbf{w})\mathbf{u}_2 = (\mathbf{u}_2^t\mathbf{w})\mathbf{u}_2 = \mathbf{u}_2(\mathbf{u}_2^t\mathbf{w}) = (\mathbf{u}_2\mathbf{u}_2^t)\mathbf{w} \ .$$

We can combine this with (3.2) to write

$$M = I - 2(\mathbf{u}_2\mathbf{u}_2^t) \ . \tag{3.3}$$

To write $\mathbf{u}\mathbf{u}^t$ down it may help to recall that for any matrix A, $A_{i,j} = \mathbf{e}_i \cdot A\mathbf{e}_j$. Applying this to $A = \mathbf{u}\mathbf{u}^t$, we find $\mathbf{e}_i \cdot (\mathbf{u}\mathbf{u}^t)\mathbf{e}_j = (\mathbf{e}_i \cdot \mathbf{u})(\mathbf{u} \cdot \mathbf{e}_j) = u_i u_j$. That is,

$$(\mathbf{u}\mathbf{u}^t)_{i,j} = u_i u_j \ .$$

Example 1 (Finding a Householder reflection in $I\!R^2$) Let $\mathbf{x} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ and let $\mathbf{y} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$. As you can easily compute, both of these vectors have the same length. In this example, $\mathbf{y} - \mathbf{x} = \begin{bmatrix} -2 \\ 4 \end{bmatrix}$, and $|\mathbf{y} - \mathbf{x}| = 2\sqrt{5}$, and so

$$\mathbf{u}_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \ .$$

Then

$$\mathbf{u}_2\mathbf{u}_2^t = \frac{1}{5}\begin{bmatrix} -1 \\ 2 \end{bmatrix} [-1, 2] = \frac{1}{5}\begin{bmatrix} 1 & -2 \\ -2 & 4 \end{bmatrix} \ .$$

Finally

$$M = I - 2\mathbf{u}_2\mathbf{u}_2^t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{5}\begin{bmatrix} 1 & -2 \\ -2 & 4 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 3 & 4 \\ 4 & -3 \end{bmatrix} \ .$$

You can now easily check that this is a reflection matrix: $M^tM = I$ and $\det(M) = -1$.

3.2 The $n \times n$ case

The formula (3.3) that we deduced in $I\!R^2$ is actually valid in $I\!R^n$ for all $n \geq 2$. From a geometric point of view, this is because any two vectors in $I\!R^n$ that are not colinear lie in a plane. If you think of the diagram above as being drawn in that plane, you see how this works.

Alternately, let $\mathbf{x}$ be any vector in $I\!R^n$. Let $\mathbf{u}$ be any unit vector. Recall the decomposition $\mathbf{x} = \mathbf{x}_\| + \mathbf{x}_\perp$ into the components of $\mathbf{x}$ that are, respectively, parallel and perpendicular to $\mathbf{u}$. Then

$$\begin{aligned} M\mathbf{x} &= M(\mathbf{x}_\| + \mathbf{x}_\perp) \\ &= M\mathbf{x}_\| + M\mathbf{x}_\perp \end{aligned} \tag{3.4}$$

Next, recall that $\mathbf{x}_\| = (\mathbf{x} \cdot \mathbf{u})\mathbf{u}$ so that

$$(\mathbf{u}\mathbf{u}^t)\mathbf{x}_\| = (\mathbf{x}_\| \cdot \mathbf{u})\mathbf{u} = (\mathbf{x} \cdot \mathbf{u})(\mathbf{u} \cdot \mathbf{u})\mathbf{u} = \mathbf{x}_\| \ .$$

Also,

$$(\mathbf{u}\mathbf{u}^t)\mathbf{x}_\perp = (\mathbf{x}_\perp \cdot \mathbf{u})\mathbf{u} = 0 \ .$$

From (3.3) we then have $M\mathbf{x}_\| = \mathbf{x}_\| - 2\mathbf{x}_\| = -\mathbf{x}_\|$ and and $M\mathbf{x}_\perp = \mathbf{x}_\perp$. Combining these computations with (3.4), we have

$$M\mathbf{x} = \mathbf{x}_\perp - \mathbf{x}_\| \ . \tag{3.5}$$

This is what a reflection does – changes the sign of the parallel component.

Now let $\mathbf{x}$ and $\mathbf{y}$ be any two vectors in $I\!R^n$ of the same length, and assume that they do not lie on the same line through the origin. As before, let $\mathbf{u} = \dfrac{1}{|\mathbf{y} - \mathbf{x}|}(\mathbf{y} - \mathbf{x})$.

We claim that the decomposition $\mathbf{x} = \mathbf{x}_\| + \mathbf{x}_\perp$ is given by

$$\mathbf{x}_\| = \frac{1}{2}(\mathbf{x} - \mathbf{y}) \qquad \text{and} \qquad \mathbf{x}_\| = \frac{1}{2}(\mathbf{x} + \mathbf{y}) \ . \tag{3.6}$$

First, the vectors in (3.6) are orthogonal by (3.1). Also it is clear that

$$\mathbf{x} = \frac{1}{2}(\mathbf{x} + \mathbf{y}) + \frac{1}{2}(\mathbf{x} - \mathbf{y}) \ . \tag{3.7}$$

Since the decomposition into parallel and perpendicular components is unique, (3.6) is true.

Then with M given by (3.3), we have from (3.5) that

$$M\mathbf{x} = \frac{1}{2}(\mathbf{x} + \mathbf{y}) - \frac{1}{2}(\mathbf{x} - \mathbf{y}) = \mathbf{y} \ .$$

That is, M reflects $\mathbf{x}$ onto $\mathbf{y}$.

Now since M is a refelction, it should be the case that M is an orthognal matrix. That is, we should have $M^tM = I$. But from the fact that $(AB)^t = B^tA^t$, $(I-2\mathbf{u}\mathbf{u}^t)^t = (I-2\mathbf{u}\mathbf{u}^t)$, so $M = M^t$. Next,

$$M^2 = (I - 2\mathbf{u}\mathbf{u}^t)^2 = I - 4\mathbf{u}\mathbf{u}^t + 4(\mathbf{u}^t\mathbf{u}^t)^2 = I \ .$$

Hence M is its own inverse. Since $M = M^t = M^{-1}$, M is clearly an orthogonal matrix, as it must be, if it is to describe a refelction in $I\!R^n$.

We have proved the following theorem:

Theorem 1 (Householder reflections) *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be any two vectors in* $I\!R^n$ *of the same length, and assume that they do not lie on the same line through the origin. Form the vector*

$$\mathbf{u} = \frac{1}{|\mathbf{y}-\mathbf{x}|}(\mathbf{y}-\mathbf{x}) \ , \tag{3.8}$$

and the matrix

$$M = I - 2\mathbf{u}\mathbf{u}^t \ . \tag{3.9}$$

Then M *is an orthogonal matrix such that* $M\mathbf{x} = \mathbf{y}$, *and*

$$.M = M^t = M^{-1} \ . \tag{3.10}$$

Example 2 (Finding a Householder reflection in $I\!R^3$) Let $\mathbf{x} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$ and let $\mathbf{y} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$. As you can easily compute, both of these vectors have the same length. Using (3.8) to compute $\mathbf{u}$, we find $\mathbf{u} = \frac{1}{\sqrt{6}}\begin{bmatrix} 1 \\ -1 \\ -2 \end{bmatrix}$. From here

$$\mathbf{u}\mathbf{u}^t = \frac{1}{6}\begin{bmatrix} 1 \\ -1 \\ -2 \end{bmatrix}[1,-1,-2] = \frac{1}{6}\begin{bmatrix} 1 & -1 & -2 \\ -1 & 1 & 2 \\ -2 & 2 & 4 \end{bmatrix}$$

and then from (3.9),

$$M = \frac{1}{3}\left(\begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix} - \begin{bmatrix} 1 & -1 & -2 \\ -1 & 1 & 2 \\ -2 & 2 & 4 \end{bmatrix}\right) = \frac{1}{3}\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & -1 \end{bmatrix} \ .$$

We now claim that if $\mathbf{v}$ is any unit vector in $I\!R^n$, and M is the Householder reflection matrix that carries $\mathbf{v}$ onto $\mathbf{e}_1$, then M is an orthogonal (hence unitary) matrix whose first column is $\mathbf{v}$.

The key fact is in (3.10): *A Householder reflection is its own inverse.* Therefore, if $M\mathbf{v} = \mathbf{e}_1$, then multiplying both sides by M, $M^2\mathbf{v} = M\mathbf{e}_1$. Since $M^2 = I$,

$$M\mathbf{e}_1 = \mathbf{v} \ .$$

But $M\mathbf{e}_1$ is the first column of M, and M is orthogonal. This is worth recording as a theorem.

Theorem 2 (Orthogonal matrices with given first column) *Let $\mathbf{v}$ be any unit vector in C^n, and let M be the Householder reflection that carries $\mathbf{v}$ to $\mathbf{e}_1$. Then M is orthogonal, and the first column of M is $\mathbf{v}$.*

Theorem 2 gives us our algorithm for writing down a unitary matrix whose first column is a given unit vector $\mathbf{v}$ in the case that $\mathbf{v}$ is real. It is not hard to adapt our formulas to the case of complex vectors. But before going on, it is worth noting another application of Householder reflection matrices in the real case.

3.3 Householder reflection matrices and the QR factorization.

Let $A = \begin{bmatrix} 3 & 5 \\ 4 & 2 \end{bmatrix}$. The first column $\mathbf{v}_1$ of this matrix is the vector $\mathbf{x}$ considered in Example 1. Let M be the Householder reflection computed in Example 1. Then with $A = [\mathbf{v}_1, \mathbf{v}_2]$ we have that $M\mathbf{v}_1 = 5\mathbf{e}_1$, and we compute

$$M\mathbf{v}_2 = \frac{1}{5}\begin{bmatrix} 23 \\ 26 \end{bmatrix} .$$

Hence, from $MA = M[\mathbf{v}_1, \mathbf{v}_2] = [M\mathbf{v}_1, M\mathbf{v}_2]$, we have

$$MA = \frac{1}{5}\begin{bmatrix} 25 & 23 \\ 0 & 26 \end{bmatrix} .$$

Let R denote the right hand side. That is,

$$R = \frac{1}{5}\begin{bmatrix} 25 & 23 \\ 0 & 26 \end{bmatrix} .$$

Since M is orthogonal, and is its own inverse, $MA = R$ is the same as

$$A = MR .$$

We have factored A into the product of an orthogonal matrix M, and an upper triangular matrix R. This is the QR decomposition of A, and Householder's idea provides and alternative to the Gramm-Schmidt orthogonalization method for producing such a factorization. The QR factorization is useful for many other purposes. Therefore, it is well worth studying in its own right.

We can use Householder reflections in $I\!R^n$ to compute a QR factorization of any matrix A. The procedure is to multiply A on the left by a sequence of Householder reflections so that the resulting product is in row reduced form. We do this one column and row at a time. Here is how it goes for an $m \times n$ matrix A:

Focus first on the first column, and find a Householder reflection matrix M_1 so that the first column of M_1A is a multiple of $\mathbf{e}_1$. Theorem 1 tells us how to do this.

At this stage the first column and row of M_1A are in good shape. You might be lucky and the rest of M_1A might be in row reduced form. If so, stop. Otherwise, *cross out the first column and row of* M_1A. What is left is an $(m-1)\times(n-1)$ matrix that we will call A_2.

Next, find a Householder matrix $\tilde{M}_2$ so that the first column of $\tilde{M}_2A_2$ is a multiple of $\mathbf{e}_1$. Now, $\tilde{M}_2$ is an $(m-1)\times(m-1)$ matrix, and we *promote* it to an $n\times n$ matrix by overwriting the lower right $(m-1)\times(m-1)$ block of $I_{n\times n}$ with $\tilde{M}_2$. Call this matrix M_2. If you multiply out M_2M_1A, you will see that the first two collumns and rows are in good shape. If M_2M_1A is in row reduced form, stop. Otherwise, cross out the first two rows and columns and go to work on what is left.

We can iterate this procedure to produce our sequence of reflections. The iteration will stop in no more than $\min\{m,n\}$ steps when we run out of either columns or rows.

Example 3 (Householder reflections and QR factorization) Let $A = \begin{bmatrix} 2 & 3 \\ 1 & 3 \\ 2 & 3 \end{bmatrix}$. Let M_1 be the Householder reflection form Example 2 so that $M_1\begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = 3\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$. We compute

$$M_1\begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & -1 \end{bmatrix}\begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ -1 \end{bmatrix}$$

Hence

$$M_1A = \begin{bmatrix} 3 & 5 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} .$$

Crossing out the first column and row, we are left with with the 2×1 matrix

$$A_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} .$$

Therefore, let $\tilde{M}_2$ be the Householder reflection that reflects $\mathbf{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ into $\mathbf{y} = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}$. To find the matrix $\tilde{M}_2$, we compute, using (3.8),

$$\mathbf{u} = \frac{1}{\sqrt{4-2\sqrt{2}}}\begin{bmatrix} \sqrt{2}-1 \\ 1 \end{bmatrix} .$$

Then,

$$\mathbf{u}\mathbf{u}^t = \frac{1}{4-2\sqrt{2}}\begin{bmatrix} 3-2\sqrt{2} & \sqrt{2}-1 \\ \sqrt{2}-1 & 1 \end{bmatrix}$$

Then from (3.9),

$$\begin{aligned} \tilde{M}_2 &= I - \frac{1}{2-\sqrt{2}}\begin{bmatrix} 3-2\sqrt{2} & \sqrt{2}-1 \\ \sqrt{2}-1 & 1 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} . \end{aligned}$$

We now "promote" $\tilde{M}_2$ to the 3×3 matrix M_2 by overwriting the lower right block of $I_{3\times 3}$ with $\tilde{M}_2$. this gives us

$$M_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & -1 \end{bmatrix} .$$

Since the first row of M_2 is the first row of the identity matrix, multiplying a matrix on the left by M_2 does not change the first row. Also, you can see that M_2 leaves any multiple of $\mathbf{e}_1$ fixed, and so M_2 leaves the first column of $M_1 A$ alone. This is good, because it is already just where we want it. Also, by construction, applying M_2 to the second column of $M_1 A$ amount to applying $\tilde{M}_2$ to bottom two entries in this collumn. that is:

$$M_2 \begin{bmatrix} 5 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 5 \\ \sqrt{2} \\ 0 \end{bmatrix} .$$

Therefore,

$$M_2 M_1 A = \begin{bmatrix} 3 & 5 \\ 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix} .$$

Let Q denote $(M_2 M_1)^t = M_1 M_2$ – since Householder reflections are symmetric – and let R denote $M_2 M_1 A$. Then Q, being the product of orthogonal matrices is itself orthogonal, and R is in row reduced form, and we have

$$A = QR .$$

This is a QR factorization of A

The procedure we used in Example 3 can be used in general: Let A be any $m \times n$ matrix $A = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$. Using the procedure described above, choose the sequence $M_1, M_2, \ldots, M_k$ of Householder reflection matrices so that $M_k \cdots M_1 M_1 A$ is in row reduced form. Define

$$Q = M_1 M_2 \cdots M_{k-1} M_k \qquad \text{and} \qquad R = M_k M_{k-1} \cdots M_2 M_1 A .$$

Then Q is orthogonal and R is in row reduced form, and $A = QR$. This gives us a QR factorization of A.

Before stating the algorithm, we mention one important point. Let $\mathbf{x}$ be any given vector in $\mathbb{R}^m$, As noted above, if we wish to reflect it onto a multiple of $\mathbf{e}_1$, we have two choices: We can reflect it onto $|\mathbf{x}|\mathbf{e}_1$ or $-|\mathbf{x}|\mathbf{e}_1$. Does it matter which one we choose?

The answer is "yes". In forming the unit vector $\mathbf{u} = (|\mathbf{y} - \mathbf{x}|)^{-1}(\mathbf{y} - \mathbf{x})$, we divide by $|\mathbf{y} - \mathbf{x}|$, and it is best to avoid a choice that makes this quantity small. If we choose*

$$\mathbf{y} = -\text{sgn}(x_1)|\mathbf{x}|\mathbf{e}_1 \tag{3.11}$$

then

$$|\mathbf{y} - \mathbf{x}|^2 = |\mathbf{x}|^2 + |\mathbf{x}|^2 + 2|x_1||\mathbf{x}| \geq 2|\mathbf{x}|^2 .$$

In what follows, when we refer to the "good" Householder reflection that reflects $\mathbf{x}$ onto a multiple of $\mathbf{e}_1$, we mean the one where the multiple is given by (3.11).

* The function $\text{sgn}(t)$ is defined by $\text{sgn}(t) = 1$ if $t \geq 0$, and $\text{sgn}(t) = -1$ otherwise.

The Householder reflection algorithm for the QR decomposition

(1) Declare two matrix variables: An $m \times n$ matrix R and an $m \times m$ matrix Q, and an integer variable r, and Initialize them as follows:

$$R \leftarrow A \qquad \text{and} \qquad Q \leftarrow I_{m\times m} \ .$$

(2) Then, while R is not in row reduced form, do the following: If the first row of R is zero, cross it out and overwrite R with the new matrix.

Otherwise, if the first row of R is non zero, compute the good Householder reflection matrix $\tilde{M}$ that reflects this column onto a multiple of $\mathbf{e}_1$. Let $\tilde{R}$ be the matrix obtained by crossing out the first row and column of $\tilde{M}R$. Promote $\tilde{M}$ to an $m \times m$ matrix M, if need be, by overwriting the lower right block of $I_{m\times m}$ by $\tilde{M}$. Update as follows:

$$R \leftarrow \tilde{R} \qquad \text{and} \qquad Q \leftarrow QM \ .$$

(3) The while loop terminates when we have run out of rows or columns, which happens in no more than n steps. When it does, return Q. This is the Q matrix so that $Q^tA = R$ is in row reduced form.

This algorithm returns the orthogonal matrix Q. If you stored the original A, you could return R too by computing $R = Q^tA$. A better thing to do would be to store columns and rows of R as you "peel them off" in the algorithm. This is easy to do with one more matrix variable.

If we apply this algorithm to an $m \times n$ matrix whose rank is r with $r < m$, then R will have $m - r$ rows at the bottom that are all zero. Therefore, when QR is multiplied out, the rightmost $m - r$ rows of Q do not actually enter the computations – they all get zero multiples in forming the product. For this reason, it is often simplest to discard them.

Define $\hat{R}$ to be the $r \times n$ matrix obtained by deleting the bottom $m-r$ rows of R. Also, define $\hat{Q}$ to be the $m \times r$ matrix obtained by deleting therightmost $m - r$ column of Q. By what we have just explained,

$$A = QR = \hat{Q}\hat{R} \ . \tag{3.12}$$

Moreover, both $\hat{Q}$ and $\hat{R}$ have rank r. Q is no longer orthogonal if $r < m$, but it still has orthnormal columns, and so it is still an isometry. That is, $\hat{Q}^t\hat{Q} = I_{r\times r}$, and $\hat{Q}\hat{Q}^t$ is the orthogonal projection onto $\text{Img}(\hat{Q})$. Now, since the ranks of both $\hat{Q}$ and A have rank r, $\text{Img}(\hat{Q})$ and $\text{Img}(A)$ have the same dimension. However, from (3.12), $\text{Img}(\hat{Q}) \subset \text{Img}(A)$, and then by the dimension principle $\text{Img}(\hat{Q}) = \text{Img}(A)$. In other words, $\hat{Q}\hat{Q}^t$ is the orthogonal projection onto $\text{Img}(A)$.

Definition (Reduced QR factorization) Let A be an $m \times n$ matrix with rank r. Let $\hat{Q}$ be an $m \times r$ isometry, and let $\hat{R}$ be an $r \times n$ matrix in row reduced form such that

$$A = \hat{Q}\hat{R} \ .$$

then this is a *reduced QR factorization of A.*

We see from the algorithm above that every $m \times n$ matrix A has a reduced QR factorization – just run the algorithm, and delete columns from Q and rows from R, as described above.

It is the reduced QR decomposition that you would want to use if you want to compute an orthogonal projection onto the image of A. In fact, it is usually the reduced QR decomposition that you will find useful in applications. All the same, it is worth noting that if $A = QR$ is a full QR factorization of A, and A has rank $r < m$, the rightmost $m-r$ columns of Q are also interesting – they are a basis for the orthogonal complement of the image of A, which is the same as the kerenl of A^t.

3.4 The complex case

So far we have dealt with vectors in $I\!R^n$. What about complex vectors? The easy generalization we made from reflections in $I\!R^2$ to reflections in $I\!R^n$ is encouraging. Perhaps the generalization to complex vectors is easy too? This is indeed the case – there is just one twist.

To generalize the fundamental formula (3.3), let $\mathbf{u}$ be any unit vector in C^n. Regard it as an $n \times 1$ matrix, and let $\mathbf{u}^*$ denote its adjoint. Then

$$U = I - 2\mathbf{u}\mathbf{u}^*$$

is a self adjoint unitary matrix.

The fact that it is self adjoint is clear. To see that it is unitary, we just compute

$$U^*U = U^2 = (I - 2\mathbf{u}\mathbf{u}^*)(I - 2\mathbf{u}\mathbf{u}^*) = I - 4\mathbf{u}\mathbf{u}^* + 4(\mathbf{u}\mathbf{u}^*)^2 \ .$$

But

$$(\mathbf{u}\mathbf{u}^*)^2 = \mathbf{u}(\mathbf{u}^*\mathbf{u})\mathbf{u}^* = \mathbf{u}I\mathbf{u}^* = \mathbf{u}\mathbf{u}^* \ .$$

Therefore, $I - 4\mathbf{u}\mathbf{u}^* + 4(\mathbf{u}\mathbf{u}^*)^2 = I - 4\mathbf{u}\mathbf{u}^* + 4\mathbf{u}\mathbf{u}^* = I$, and so

$$U^*U = I \ .$$

Next, suppose that $\mathbf{z}$ and $\mathbf{w}$ are two vectors in C^n with the same length. Replacing $\mathbf{w}$ by $e^{i\theta}\mathbf{w}$ for some θ, we can assume that $\langle \mathbf{z}, \mathbf{w} \rangle$ is a real number, so that $\langle \mathbf{z}, \mathbf{w} \rangle = \langle \mathbf{w}, \mathbf{w} \rangle$. Having made this replacement, define $\mathbf{u}$ by

$$\mathbf{u} = \frac{1}{|\mathbf{w} - \mathbf{z}|}(\mathbf{w} - \mathbf{z}) \ .$$

Then, because $\langle \mathbf{z}, \mathbf{w} \rangle = \langle \mathbf{w}, \mathbf{w} \rangle$, $|\mathbf{w} - \mathbf{z}|^2 = 2|\mathbf{z}|^2 - 2\langle \mathbf{w}, \mathbf{z} \rangle$, and so

$$\begin{aligned} U\mathbf{z} &= \mathbf{z} - \frac{2}{|\mathbf{w} - \mathbf{z}|^2}(\langle \mathbf{w}, \mathbf{z} \rangle - |\mathbf{z}|^2)(\mathbf{w} - \mathbf{z}) \\ &= \mathbf{z} - \frac{2}{2|\mathbf{z}|^2 - 2\langle \mathbf{w}, \mathbf{z} \rangle}(\langle \mathbf{w}, \mathbf{z} \rangle - |\mathbf{z}|^2)(\mathbf{w} - \mathbf{z}) \\ &= \mathbf{z} - (\mathbf{z} - \mathbf{w}) \\ &= \mathbf{w} \ . \end{aligned}$$

Hence our formulas for Householder reflections generalize easily from $I\!R^n$ to C^n. All that we have to do is to replace transposes by adjoints, and adjust the "phase" $\epsilon^{i\theta}$ of $\mathbf{w}$.

Applying unitary Householder reflections in C^n in the same manner as one applies orthogonal Householder reflections in $I\!R^n$, one can factor any complex $n \times n$ matrix A as

$$A = QR$$

where Q is unitary, and R is upper triangular. This is the complex version of the QR factorization. If A has all real entries, then we can do this with Q orthogonal, which is a special case, but when A is complex, we must in general allow Q to be unitary instead. We can now easily adapt the QR iteration to complex matrices. In the next section, we shall see how this can be applied to reveal the complex eigenvalues of an $n \times n$ matrix.

Problems

3.1 Let $\mathbf{x} = \begin{bmatrix} 11 \\ 2 \end{bmatrix}$ Find both reflection matrices M such that $M\mathbf{x}$ is a multiple of $\mathbf{e}_1$.

3.2 Let $\mathbf{x} = \begin{bmatrix} 5 \\ 2 \\ 4 \\ 2 \end{bmatrix}$ Find both reflection matrices M such that $M\mathbf{x}$ is a multiple of $\mathbf{e}_1$.

3.3 Let $\mathbf{x} = \begin{bmatrix} 5 \\ 1 \\ 2 \\ 3 \\ 5 \end{bmatrix}$ Find both reflection matrices M such that $M\mathbf{x}$ is a multiple of $\mathbf{e}_1$.

3.4 Let $A = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 2 & 1 \\ 2 & -4 & 1 \end{bmatrix}$. Compute a QR factorization of A using the Householder reflection method. separately list all of the Householder reflections used along the way.

3.5 Let $A = \begin{bmatrix} 3 & 3 & 3 \\ 0 & 1 & 1 \\ 4 & 4 & 1 \end{bmatrix}$. Compute a QR factorization of A using the Householder reflection method. separately list all of the Householder reflections used along the way.

3.6 Let $A = \begin{bmatrix} 2 & -2 & 3 & 1 \\ 1 & 3 & 1 & 2 \\ 2 & 2 & 1 & 1 \end{bmatrix}$. Compute a QR factorization of A using the Householder reflection method. separately list all of the Householder reflections used along the way.

3.7 Let $\mathbf{z} = \begin{bmatrix} 1+i \\ -3+2i \\ 1-4i \end{bmatrix}$. Compute the unitary Householder reflection that reflects $\mathbf{z}$ onto a multiple of $\mathbf{e}_1$. (The multiple will have to be chosen so that it results in a vector $\mathbf{w}$ that is the same length as $\mathbf{z}$, and such that $\langle \mathbf{z}, \mathbf{w} \rangle$ is a real number).

3.8 Let $\mathbf{z} = \begin{bmatrix} 5-i \\ i \\ 5 \\ 1-3i \end{bmatrix}$. Compute the unitary Householder reflection that reflects $\mathbf{z}$ onto a multiple of $\mathbf{e}_1$. (The multiple will have to be chosen so that it results in a vector $\mathbf{w}$ that is the same length as $\mathbf{z}$, and such that $\langle \mathbf{z}, \mathbf{w} \rangle$ is a real number).

Section 4: QR Iteration

4.1 What the QR iteration is

In this section, we present an algorithm for computing one eigenvector of of any square matrix A. This algorithm is based on the QR factorization.

Here is a strategy for finding the eigenvalues of an $n\times n$ matrix A: Find a similar matrix B whose eigenvalues are easy to determine. Since A and B are similar, they have the same eigenvalues, and so from here, the eigenvalues of A are easily determined.

There is a way of implementing this strategy using the QR factorization. Let A be any $n \times n$ matrix, and let $A = QR$ be a QR factorization of A. Then, since Q is orthogonal, $Q^{-1} = Q^t$, and so

$$Q^tAQ = Q^t(QR)Q = (Q^tQ)RQ = RQ \ .$$

In other words, QR and RQ are similar matrices, and so they have the same eigenvalues.

Let $A^{(1)}$ denote the new matrix RQ. Not only is $A^{(1)}$ similar to A, but we can get new similar matrix by repeating the procedure. Let $A^{(1)} = Q^{(2)}R^{(2)}$ be a QR factorization of $A^{(1)}$, and define

$$A^{(2)} = R^{(2)}Q^{(2)} \ .$$

By what we saw above, $A^{(2)}$ is similar to $A^{(1)}$, and hence to A. We can keep going in this way, and produce an infinite sequence of matrices, all similar to A.

Here is the good news: It turns out that in many cases, this sequence will rapidly converge to an upper triangular matrix T. The eigenvalues of an upper triangular matrix are the diagonal entries, so we can just *see* what they are. Since T is similar to A, what we are seeing are in fact the eigenvalues of A!

The QR iteration

For any $n \times n$ matrix A, define

$$A^{(0)} = A \ . \tag{4.1}$$

The recursively define the matrices $A^{(k)}$, $Q^{(k)}$ and $R^{(k)}$ by

$$Q^{(k)}R^{(k)} = A^{(k-1)} \tag{4.2}$$

and then

$$A^{(k)} = R^{(k)}Q^{(k)} \ . \tag{4.3}$$

In many cases, we will find that $\lim_{k\to\infty} A^{(k)} = T$ where T is upper triangular. Since each $A^{(k)}$ us similar to A, it follows that the eigenvalues of A are the diagonal entries of T. The QR iteration is a marvelous eigenvalue revealing algorithm.

Example 1 (Running the QR iteration) We begin with a 2×2 example. Let

$$A = \begin{bmatrix} -3 & 4 \\ -6 & 7 \end{bmatrix} \ .$$

It is easy to compute the eigenvalues and eigenvectors of A. The characteristic polynomial $p(t) = \det(A-tI)$ is

$$p(t) = t^2 - 4t + 3$$

which means that the eigenvalues are 1 and 3.

We now report the results of running the QR iteration. The computations are easily done using a computer program that computes QR factorizations. One finds:

$$\begin{aligned}
A^{(1)} &= \frac{1}{5}\begin{bmatrix} 12 & 48 \\ -2 & -1 \end{bmatrix} \\
A^{(2)} &= \frac{1}{89}\begin{bmatrix} 293 & 884 \\ -6 & -63 \end{bmatrix} \\
A^{(3)} &= \frac{1}{965}\begin{bmatrix} 2981 & 9632 \\ -18 & 897 \end{bmatrix}
\end{aligned}$$

After here, the fractions get messy, so we report the results in decimal form. Keeping 6 digits, we have

$$A^{(10)} = \begin{bmatrix} 3.00004 & 9.99999 \\ .781649 \times 10^{-5} & .999961 \end{bmatrix} .$$

From the frist 10 terms you may well guess that

$$\lim_{k\to\infty} A^{(k)} = \begin{bmatrix} 3 & 10 \\ 0 & 1 \end{bmatrix} .$$

This would imply that A is similar to the matrix on the right. Is this the case? First, the eigenvalues are 3 and 1, which must be the case if there is to be similarity.

To see that there is similarity, let $S = \begin{bmatrix} 2 & -3 \\ 3 & 2 \end{bmatrix}$. We will explain in Section 3 where S came from, but you can now easily compute that

$$S^{-1}AS = \begin{bmatrix} 3 & 10 \\ 0 & 1 \end{bmatrix} ,$$

so that A is indeed similar to the upper triangular matrix $\begin{bmatrix} 3 & 10 \\ 0 & 1 \end{bmatrix}$. Now of course, using the QR iteration is not at all an efficient way to reveal the eigenvalues of a 2×2 matrix. But now that we have seen what it does here, let's try it out on a larger matrix.

Example 2 (Running the QR iteration for a 4×4 matrix) Let's build a 4×4 matrix with known eigenvalues. Let

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} .$$

Now let V be any 4×4 matrix with a "nice" inverse. For example, take

$$V = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 1 & 3 & 1 & 2 \\ 1 & 1 & 2 & 1 \\ 1 & 2 & 3 & 1 \end{bmatrix} .$$

Then

$$V^{-1} = \begin{bmatrix} -4 & 3 & 5 & -3 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 \\ 3 & -2 & -2 & 1 \end{bmatrix} .$$

Now let $A = VDV^{-1}$. This gives us

$$A = \begin{bmatrix} 19 & -12 & -14 & 8 \\ 17 & -10 & -14 & 8 \\ 12 & -9 & -9 & 7 \\ 13 & -10 & -12 & 10 \end{bmatrix} .$$

From the way we cooked this matrix up, we know its eigenvalues: $1, 2, 3$ and 4. But if we covered out tracks, this wouldn't have been so obvious. Nonetheless, the QR iteration reveals these eigenvalues.

If you compute, in decimal form, the sequence $\{A^{(k)}\}$, you find, for example, that to 6 decimal places,

$$A^{(20)} = \begin{bmatrix} 3.99905 & 1.08347 & 8.31213 & -45.6654 \\ .000867 & 3.00101 & -.527802 & 9.76837 \\ 0 & .000106 & 1.99994 & -.841663 \\ 0 & 0 & 0 & 1.00000 \end{bmatrix} .$$

This suggets that $\lim_{k\to\infty} A^{(k)}$ is an upper triangular matrix with diagonal entries 4, 3, 2 and 1. This is the case as we shall see. Once again, the QR iteration has revealed the eigenvalues. Notice also that they appear on the diagonal in order of decreasing size. This too, as we shall see, is not an accident.

We have just seen two examples in which the QR iterations reveals the eigenvalues of a sqaure matrix. Used in this simple form, it does not always work. Here is an example.

Example 3 (The QR iteration spinning its wheels) Let $D = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ and let $V = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}$, and let

$$A = VDV^{-1} = \begin{bmatrix} 5 & -4 \\ 6 & -5 \end{bmatrix} .$$

Now let's run the QR iteration we find

$$A^{(1)} = \frac{1}{61} \begin{bmatrix} 5 & 616 \\ 6 & -5 \end{bmatrix} .$$

At the next stage, something remarkable happens:

$$A^{(2)} = \begin{bmatrix} 5 & -4 \\ 6 & -5 \end{bmatrix} .$$

We are back to where we started! It is clear that for this matrix, the QR iteration will not ever converge. Instead, it will just cycle back and forth between the two matrices $\begin{bmatrix} 5 & -4 \\ 6 & -5 \end{bmatrix}$ and $\frac{1}{61}\begin{bmatrix} 5 & 616 \\ 6 & -5 \end{bmatrix}$.

Why did this happen? As we shall see, it happened because the eigenvalues of A, 1 and -1, have the same absolute value.

We close this subsection by stating the QR algorithm in a slightly provisional form. The provisional aspect is that we have not yet analyzed the algorithm to understand when it will work, and how long that might take if it does. Therefore, we do not yet possess a good stopping rule. In general, even when things are working well, the while loop in the algorithm as formulated below would run forever.

The QR algorithm

We are given an $n \times n$ matrix A as input. Declare two $n \times n$ matrix variables B and U. Initialize them as follows:

$$B \leftarrow A \qquad \text{and} \qquad U \leftarrow I \ .$$

Then, while B is not upper triangular:

(1) Compute a QR factorization $B = QR$ of B.

(2) Update the variables as follows:

$$B \leftarrow RQ \qquad \text{and} \qquad U \leftarrow UQ \ .$$

Let $A(k)$ be the matrix B at the kth step of the iteration. As we have explained above, for each k $A^{(k)}$ is similar to $A^{(k-1)}$, and so for each k, $A^{(k)}$ is similar to A. Hence if the algorithm does terminate in an upper triangular matrix, the diagonal entries of that matrix are the eigenvalues of A.

You may well be wondering why we are keeping track of the matrix U, wich is a cummulative record of the orthogonal matrices Q used in the successive QR factorizations.

There is a good reason: Let $U^{(k)}$ be the matrix U at the kth step of the iteration. This is the cummulative record of the similarity transformations used up to the kth step, so it is not hard to see that

$$\left(U^{(k)}\right)^{-1} AU^{(k)} = A^{(k)} \ . \tag{4.4}$$

We will give a formal proof of this n the next subsection, but you can probably see why this is so on the basis if the examples we have done.

Therefore suppose that the algorithm does terminate in an upper triangular matrix T. Let U be the final value of the U matrix. Then (4.4) becomes

$$U^{-1}AU = T \ , \tag{4.5}$$

or equivalently,

$$A = UTU^{-1} \ . \tag{4.6}$$

What we have worked out is a Schur decomposition of A.

4.2 When and how QR iteration works

The QR iteration does not always “work”, as we have seen in Example 3. But if we can understand how it works when it does, we can hope to modify it, and male it work in general. We can also hope to make it work faster.

The key to understanding the QR iteration lies with two other sequences of matrices. Let $Q^{(k)}$ and $R^{(k)}$ be the matrices figuring in the QR decomposition at the kth stage. Define two other sequences of matrices as follows:

$$U^{(k)} = Q^{(1)}Q^{(2)} \cdots Q^{(k)} \ , \tag{4.7}$$

and

$$T^{(k)} = R^{(k)} R^{(k-1)} \cdots R^{(1)} \ . \tag{4.8}$$

We have already met the sequence $U^{(k)}$ in the last section. Since any product of orthogonal matrices is orthogonal, $U^{(k)}$ is orthogonal for every k. Likewise, any product of upper tiriangular matrices is upper triangular, $T^{(k)}$ is upper triangular for every k.

It turns out that $U^{(k)}T^{(k)}$ is the QR factorization of A^k, the kth power of A. This fact that makes the QR algorithm work, when it does.

Theorem 1 (QR iteration and powers) *Let A be any $n \times n$ matrix, and let $A^{(k)}$, $U^{(k)}$ and $T^{(k)}$ be the sequences of matrices specified in (4.1) through (4.8). Then*

$$A^k = U^{(k)} T^{(k)} \ , \tag{4.9}$$

and

$$A U^{(k)} = U^{(k)} A^{(k)} \ . \tag{4.10}$$

Before proving the theorem, let us note one consequence of (4.10) that will be very useful to us soon. Suppose that for some value of k, $A^{(k)}$ is upper triangular, at least to as many decimal places of accuracy as we cares about. Let μ be the upper left entry of $A^{(k)}$. Then, applying $A^{(k)}$ to $\mathbf{e}_1$ is very simple:

$$A^{(k)}\mathbf{e}_1 = \mu \mathbf{e}_1 \ .$$

Combining this with (4.10),

$$A U^{(k)} \mathbf{e}_1 = U^{(k)} A^{(k)} \mathbf{e}_1 = \mu U^{(k)} \mathbf{e}_1 \ . \tag{4.11}$$

That is, if we define $\mathbf{q} = U^{(k)}\mathbf{e}_1$, which is the first column of $U^{(k)}$, we have

$$A\mathbf{q} = \mu \mathbf{q} \ ,$$

and the first column of $U^{(k)}$ is an approximate eigenvector of A!

The approximation is accurate to the extent that that first column of $A^{(k)}$ is a multiple of $\mathbf{e}_1$. Thus, the QR iteration can also be used reveal eigenvectors. It is not hard to find the others this way, but as we will explain in the next section, finding one eigenvector is the key to a recursive procedure for computing an upper triangular matrix similar to A.

Also, before doing the proof in general, lets look at a few iterations. Let $A = QR$, and determine $\tilde{Q}$ and $\tilde{R}$ by finding a QR iteration of RQ. Then we have $\tilde{Q}\tilde{R} = RQ$, and

$$A^2 = (QR)(QR) = Q(RQ)R = Q(\tilde{Q}\tilde{R})R = (Q\tilde{Q})(\tilde{R}R) \ .$$

This is the $k = 2$ case of (4.9). The following proof just uses induction to establish the general case.

Proof: First, we note that if $A = QR$, then $Q^{(1)} = U^{(1)} = Q$ and $T^{(1)} = R^{(1)} = R$, and $A^{(1)} = RQ$. Therefore, (4.9) reduces to $A = QR$, and (4.10) reduces to $A = Q(RQ)Q^t = QR(QQ^t) = QR$, so both (4.9) and (4.10) hold for $k = 1$. We now proceed by induction, and assume that both (4.9) and (4.10) hold for some value of k. Since (4.10) can be written as

$$A = U^{(k)}A^{(k)}\left(U^{(k)}\right)^t \ ,$$

we have, also using (4.9),

$$\begin{aligned}
A^{k+1} = AA^k &= \left(U^{(k)}A^{(k)}\left(U^{(k)}\right)^t\right)\left(U^{(k)}T^{(k)}\right)\\
&= U^{(k)}A^{(k)}T^{(k)}\\
&= U^{(k)}Q^{(k+1)}R^{(k+1)}T^{(k)}\\
&= U^{(k+1)}T^{(k+1)} \ .
\end{aligned}$$

This proves (4.9) for $k + 1$.

Next, (4.10) can also be written as

$$A^{(k)} = \left(U^{(k)}\right)^t AU^{(k)} \ .$$

Then from (4.2), $A^{(k)} = Q^{(k+1)}R^{(k+1)}$. Therefore,

$$Q^{(k+1)}R^{(k+1)} = \left(U^{(k)}\right)^t AU^{(k)} \ .$$

Multiplying on the left $(Q^{(k+1)})^t$ and on the right by $Q^{(k+1)}$, we have

$$\begin{aligned}
R^{(k+1)}Q^{(k+1)} &= \left(U^{(k)}Q^{(k+1)}\right)^t AU^{(k)}Q^{(k+1)}\\
&= \left(U^{(k+1)}\right)^t AU^{(k+1)} \ .
\end{aligned}$$

This proves (4.10) for $k + 1$. ■

Example 4 (QR and powers for a simple 2×2 matrix) Consider the 2×2 matrix $A = \begin{bmatrix} -3 & 4 \\ -6 & 7 \end{bmatrix}$ from Example 1. We will now apply Theorem 1 to explain the convergence to an upper triangular matrix that we observed in Example 1. This is very important to understand, but once you see how it works in this example, you will understand the key ideas in the general case. Therefore, this example is worth studying closely.

As we saw in Example 1, the matrix A has eigenvalues 1 and 3. As you can check,

$$A\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix} \qquad \text{and} \qquad A\begin{bmatrix}2\\3\end{bmatrix} = 3\begin{bmatrix}2\\3\end{bmatrix} \ .$$

Normalizing these eigenvectors, we get our basis $\{\mathbf{u}_1, \mathbf{u}_2\}$ with

$$\mathbf{u}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} \qquad \text{and} \qquad \mathbf{u}_2 = \frac{1}{\sqrt{13}}\begin{bmatrix}2\\3\end{bmatrix} \ .$$

Now let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the columns of A. You can easily solve to find that

$$\mathbf{v}_1 = (3\sqrt{2})\mathbf{u}_1 - (3\sqrt{13})\mathbf{u}_2 \qquad \text{and} \qquad \mathbf{v}_2 = -(2\sqrt{2})bu_1 + (3\sqrt{13})\mathbf{u}_2 \ .$$

Since $A^{k+1} = A^k[\mathbf{v}_1, \mathbf{v}_2]$, the columns of A^{k+1} are

$$A^k\mathbf{v}_1 = A^k((3\sqrt{2})\mathbf{u}_1 - (3\sqrt{13})\mathbf{u}_2) = (3\sqrt{2})\mathbf{u}_1 - (3\sqrt{13})3^k\mathbf{u}_2$$

and

$$A^k\mathbf{v}_2 = A^k(-(2\sqrt{2})bu_1 + (3\sqrt{13})\mathbf{u}_2) = -(2\sqrt{2})bu_1 + (3\sqrt{13})3^k\mathbf{u}_2$$

When k is large, 3^k is very large, and so we have both

$$\frac{1}{|A^k\mathbf{v}_1|}A^k\mathbf{v}_1 \approx \mathbf{u}_2$$

and

$$\frac{1}{|A^k\mathbf{v}_2|}A^k\mathbf{v}_2 \approx \mathbf{u}_2 \ .$$

Therefore, if we compute an orthonormal basis of $I\!R^2$ starting from the columns of A^{k+1} for large k, the first vector in the orthonormal basis will be pretty close to $\mathbf{u}_2$. The second vector in the basis must be orthogonal to this one, so it is pretty close to $\pm\mathbf{u}_2^\perp$.

Computing the QR factorization of A^{k+1} gives us just such a basis – an orthonormal basis of $I\!R^2$ in which the first vector is the unit vector obtained by normalizing $A^k\mathbf{v}_1$, and the second vector is a unit vector orthogonal to the first one. Hence, from (4.9), the first column of $U^{(k+1)}$ will, for large k, be pretty close to $\mathbf{u}_2$, and the second column will be some unit vector orthogonal to the first one.

Fix a large value of k, and write $U^{(k+1)} = [\mathbf{q}_1, \mathbf{q}_2]$. Then from (4.10),

$$\begin{aligned} A^{(k+1)} &= \begin{bmatrix}\mathbf{q}_1\\ \mathbf{q}_2\end{bmatrix} A[\mathbf{q}_1, \mathbf{q}_2] \\ &= \begin{bmatrix}\mathbf{q}_1\\ \mathbf{q}_2\end{bmatrix} [A\mathbf{q}_1, A\mathbf{q}_2] \\ &= \begin{bmatrix}\mathbf{q}_1\cdot A\mathbf{q}_1 & \mathbf{q}_1\cdot A\mathbf{q}_2\\ \mathbf{q}_2\cdot A\mathbf{q}_1 & \mathbf{q}_2\cdot A\mathbf{q}_2\end{bmatrix} \ . \end{aligned}$$

Now since $\mathbf{q}_1 \approx \mathbf{u}_2$ and $A\mathbf{u}_2 = 3\mathbf{u}_2$,

$$A\mathbf{q}_1 \approx 3\mathbf{q}_1$$

and so

$$\mathbf{q}_2 \cdot A\mathbf{q}_1 \approx 0 \qquad \text{and} \qquad \mathbf{q}_1 \cdot A\mathbf{q}_1 \approx 3$$

since $\{\mathbf{q}_1, \mathbf{q}_2\}$ is an orthonormal basis. Therefore, $A^{(k+1)}$ is approximately lower triangular, and as k tends to infinity, it becomes exactly so. We have already seen that the upper diagonal entry is 3 in the limit as k tends to infinity. Since $A^{(k)}$ is similar to A, it follws that the other diagonal entry must be 1, the other eigenvalue of A, in this same limit. This is exactly what we found in Example 1.

Having explained how the QR iteration worked to reveal eigenvalues in one example, we turn to a more general investigation. Make sure you have thoroughly understood Example 4 before proceeding!

Let A be an $n \times n$ matrix. We will suppose that A is diagonalizable, and we will oder the eigenvalues so that they are non increasing in absolute value. That is:

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \ .$$

To start with, lets make a stronger assumption:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| \ . \tag{4.12}$$

In particular, this assumption precludes any complex conjugate pairs of eigenvalues – because then some pair would have the same abslute value. Hence, if A has real entries, our assumption implies that all of the eigenvalues of A are real. To use (4.12) in a quantitative way, define

$$r = \max_{1 \leq j \leq n-1} \left\{ \frac{|\lambda_j|}{|\lambda_{j+1}} \right\} \ . \tag{4.13}$$

Fix a large value of k, and let $U = U^{(k)}$, so that by Theorem 1, this is the orthogonal matrix in the QR decomposition of A^k. Define the vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\}$ by

$$U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n] \ .$$

Let $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ be a basis of eigenvectors of A corresponding to the sequence of eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. Then with

$$V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n] \ ,$$

and with D being the diagonal matrix whose jth diagonal entry is λ_j, we have $A = VDV^{-1}$, and hence for any positive integer k, the kth power of A is given by

$$A^k = VD^kV^{-1} \ . \tag{4.14}$$

We are going to show that when r^k is sufficiently small, there is an invertible upper triangular matrix R with

$$U \approx VR \ . \tag{4.15}$$

From this it will follow that U^tAU is approximately upper triangular.

To see this, suppose that (4.15) were exactly true. Then, since the columns of V are eigenvectors of A, $AV = VD$. Therefore,

$$\begin{aligned}
U^t AU &= U^t A(VR) \\
&= U^t(AV)R \\
&= U^t(VD)R \\
&= (U^t)V(DR) \\
&= (U^t)UR^{-1}(DR) \\
&= (U^tU)(R^{-1}DR) \\
&= R^{-1}DR\ .
\end{aligned}$$

Now the inverse of an upper triangular matrix is also upper traingular, and diagonal matrices are of course a special kind of upper triangular matrix. Finally, the product of upper triangular matrices is upper triangular, and so $T = R^{-1}DR$ is upper triangular. This proves that if (4.15) were exactly true, it would be the case that U^tAU is upper triangular.

Therefore, to explain why the QR algorithm will always work under the assumption (4.12), we just have to show why (4.15) is true for large values of k. With R being upper triangular, what (4.12) says is that $\mathbf{u}_1$ is a multiple of $\mathbf{v}_1$, and $\mathbf{u}_2$ is a linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$ only , and that $\mathbf{u}_3$ is a linear combination of $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ only and so on. This is what we have to prove to explain the QR algorithm:

• *When k is sufficiently large, so that r^k is sufficiently small, $\mathbf{u}_1$ is a approximately multiple of $\mathbf{v}_1$, and $\mathbf{u}_2$ is approximately a linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$ only , and that $\mathbf{u}_3$ is approximately a linear combination of $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ only and so on.*

Let $\mathbf{w}_j$ be the jth row of V^{-1}. Then by (4.14),

$$A^k = VD^kV^{-1} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n] \begin{bmatrix} \lambda_1^k \mathbf{w}_1 \\ \lambda_2^k \mathbf{w}_2 \\ \vdots \\ \lambda_n^k \mathbf{w}_n \end{bmatrix} = \sum_{j=1}^{n} \lambda_j^k \mathbf{v}_j \mathbf{w}_j^t\ .$$

Now under our assumption, when k is large, $|\lambda_1|^k$ is *much* larger than $|\lambda_j|^k$ for any other vlaue of j. That is, neglecting the other terms, we have

$$A^k \approx \lambda_1^k \mathbf{v}_1 \mathbf{w}_1^t\ . \tag{4.16}$$

By (4.13), what we are throwing away is smaller by a factor of at least r^k than what we are keeping. Since $r < 1$, this is very small if k is large. If k is so large that $r^k < 10^{-6}$, the approximation in (4.16) can be expected to be accurate to about 5 decimal places.

Now let's consider computing the QR factorization of A^k, but let's think of it in terms of the Gramm–Schmidt orthonormalization procedure. According to (4.16), the columns of A^k are all very nearly multiples of $\mathbf{v}_1$. In particular, the first column of A^k is very nearly a multiple of $\mathbf{v}_1$. We get the first vector $\mathbf{u}_1$ in the orthonormalization by normalizing the first column of A^k. But to high accuracy, this is the same as just normalizing $\mathbf{v}_1$.

When we go on to compute $\mathbf{u}_2$, we build this out of the second column of A^k. In the approximation (4.16), the second column is a multiple of the first, and we get nothing. So we have to go to the next approximation:

$$A^k \approx \lambda_1^k \mathbf{v}_1 \mathbf{w}_1^t + \lambda_2^k \mathbf{v}_2 \mathbf{w}_2^t \ . \tag{4.17}$$

In this approximation, the second column is a linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$. Contributions from other columns are smaller by a factor of at least r^k. Hence we can find $\mathbf{u}_2$ by applying the Gramm–Schmidt algorithm to $\{\mathbf{v}_1, \mathbf{v}_2\}$.

Continuing in this way, we see that for each j, we can find $\mathbf{u}_j$ by applying the Gramm–Schmidt algorithm to $\{\mathbf{v}_1, \mathbf{v}_2 \dots, \mathbf{v}_j\}$. That is, there is an upper triangular matrix R so that

$$[\mathbf{u}_1, \mathbf{u}_2 \dots, \mathbf{u}_n] = [\mathbf{v}_1, \mathbf{v}_2 \dots, \mathbf{v}_n]R \ .$$

This line of reasoning shows that under the assumptions we have made above, as k tends to infinity, $A^{(k)}$ tends to an upper triangular matrix, and the eigenvalues of A are revealed on the diagonal of this limiting upper triangular matrix. The result is the following:

Theorem 2 *Let A be any $n \times n$ matrix, and suppose that A has n distinct eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ satisfying (4.12). Let $\{A^{(k)}\}$ be the sequence of matrices produced from A by the QR algorithm. Then $\lim_{n\to\infty} A^{(k)}$ exists and is an upper triangular matrix T*

Moreover, we see that the difference between $A^{(k)}$ and T tends to zero exponentialy fast – roughly like r^k. So the more distinct the magnitudes of the eigenvalues are, the better the rate of convergence.

4.3 What to do when the QR iteration bogs down

We now see what went wrong in Example 3: The eigenvalues of the matrix $A = \begin{bmatrix} 5 & -4 \\ 6 & -5 \end{bmatrix}$ are 1 and -1, and so (4.12) is not satisfied. But we can do something about this! Define a new matrix B by

$$B = A + aI \ ,$$

where a is any number. If $\mathbf{v}$ is an eigenvalue of A with eigenvalue 1, then

$$B\mathbf{v} = A\mathbf{v} + a\mathbf{v} = (1+a)\mathbf{v}$$

so that $\mathbf{v}$ is an eigenvector of B with eigenvalue $1+a$. Hence $1+a$ is an eigenvalue of B. In the same way we se that $-1+a$ is also an eigenvalue of B. That is, adding aI to A shifts the eigenvalues of A by a.

Now for any $a > 0$, $|1+a| > |-1+a|$, while for any $a < 0$, $|1+a| < |-1+a|$. Either way, Theorem 2 applies to B. Hence we can apply the QR algorithm to B, and in this way find the eigenvalues of B. But then we know the eigenvalues of A: Just subtract a off of the eigenvalues of B.

For example, let $a = 1$. Then

$$B = A - I = \begin{bmatrix} 4 & -4 \\ 6 & -6 \end{bmatrix} .$$

Then we find the QR decomposition of B to be given by

$$Q = \frac{1}{\sqrt{13}} \begin{bmatrix} 2 & 3 \\ 3 & -2 \end{bmatrix} \qquad \text{and} \qquad R = \sqrt{13} \begin{bmatrix} 2 & -2 \\ 0 & 0 \end{bmatrix} .$$

We then find

$$RQ = \begin{bmatrix} -2 & 10 \\ 0 & 0 \end{bmatrix} .$$

This is upper triangular, so for the shifted matrix B, the QR algorithm converges in one step! We see that the eigenvalues of the shifted matrix B are -2 and 0. Hence the eigenvalues of the original unshifted matrix A are -1 and 1.

The best thing about this is it really did not matter what we added to B – any non zero shift would do. So this is what to do when the QR algorithm starts spinning its wheels: Just shift and continue. If there are pairs of complex conjugate eigenvalues, it is necessary to shift by a complex number a to separate the absolute values of the eigenvalues. That means that one must use the complex version of the QR algorithm in this case, but we know how to do this.

Problems

4.1 (a) For $k = 1, 2$ and 3 compute $A^{(k)}$ for $A = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}$. Then, keeping six decimal places, compute $A^{(20)}$ numerically, and guess the eigenvalues of A.
(b) Compute the eigenvectors and eigenvalues of A.

4..2 (a) For $k = 1, 2$ and 3 compute $A^{(k)}$ for $A = \begin{bmatrix} -1 & 2 \\ 4 & -3 \end{bmatrix}$. Then, keeping six decimal places, compute $A^{(20)}$ numerically, and guess the eigenvalues of A.
(b) Compute the eigenvectors and eigenvalues of A.

4.3 (a) For $k = 1, 2$ and 3 compute $A^{(k)}$ for $A = \begin{bmatrix} -11 & -6 \\ 20 & 11 \end{bmatrix}$. Then, keeping six decimal places, compute $A^{(20)}$ numerically, and guess the eigenvalues of A.
(b) Compute the eigenvectors and eigenvalues of A.

4.4 (a) For $k = 1, 2$ and 3 compute $A^{(k)}$ for $A = \begin{bmatrix} -1 & 9 \\ 4 & -1 \end{bmatrix}$. Then, keeping six decimal places, compute $A^{(20)}$ numerically, and guess the eigenvalues of A.
(b) Compute the eigenvectors and eigenvalues of A.

4.5 (a) For $k = 1, 2$ and 3 compute $A^{(k)}$ for $A = \begin{bmatrix} -7 & 20 \\ 4 & -9 \end{bmatrix}$. Then, keeping six decimal places, compute $A^{(20)}$ numerically, and guess the eigenvalues of A.

(b) Compute the eigenvectors and eigenvalues of A.

4.6 Let $A = \begin{bmatrix} -5 & 10 & -4 & 2 \\ -24 & 19 & -4 & 12 \\ -18 & 14 & 1 & 6 \\ -4 & 10 & -4 & 1 \end{bmatrix}$. This matrix has integer eigenvalues. Run the QR iteration to reveal these eigenvalues, and then find the corresponding eigenvectors.

4.7 Let $A = \begin{bmatrix} -15 & 5 & 1 & 11 \\ -3 & 11 & 3 & -9 \\ 11 & 7 & 7 & -23 \\ -5 & -5 & 1 & 1 \end{bmatrix}$. This matrix has integer eigenvalues. Run the QR iteration to reveal these eigenvalues, and then find the corresponding eigenvectors.

4.8 Let $A = \begin{bmatrix} 9 & 1 & 5 & -17 \\ 11 & 1 & 9 & -23 \\ 5 & 5 & 5 & -17 \\ 7 & 1 & 5 & -15 \end{bmatrix}$. Check that 2, -2, 4 and -4 are eigenvalues of A. Run the QR iteration for 10 steps. Does it show any sign of converging? Can you explain what you see?

Section 5. Back to Schur factorization

5.1 Putting the pieces together

Recall our algorithm for computing a Schur factorization:

Algorithm for computing a Schur factorization

Let A be a given $n \times n$ matrix. Declare an integer variable k, and $n \times n$ matrix variable V, and a matrix variable B of adjustable size. Initialize them as follows:

$$B \leftarrow A \qquad V \leftarrow I \qquad k \leftarrow n \ .$$

Then, while $k > 1$:

(1) Find one normalized eigenvector $\mathbf{u}$ of B.

(2) Find a unitary matrix Q whose first column is a multiple of $\mathbf{u}$.

(3) Let C be the $(k-1) \times (k-1)$ matrix in the lower right of Q^*BQ, and update

$$B \leftarrow C \ .$$

(4) define an $n \times n$ unitary matrix V as follows: If $k = n$, let $V = Q$. If $k < n$, define

$$V = \begin{bmatrix} I_{(n-k)\times(n-k)} & 0 \\ 0 & Q \end{bmatrix} \ .$$

Then update according to

$$U \leftarrow UV \qquad \text{and} \qquad k - 1 \leftarrow k \ .$$

We now posses subroutines needed to implement each step. We can use the QR iteration – hopefully – to find an eigenvector, and then a appropriate Householder reflection matrix for the unitary matrix Q.

Example 1 (Computation of a Schur Factorization) Let $A = \begin{bmatrix} -13 & 10 & 8 \\ -8 & 7 & 4 \\ -4 & 10 & 9 \end{bmatrix}$. The characteristic polynomial of A is a cubic polynomial with real coefficients. As such, it has one real root, and we could determine an approximate value graphically. However, let us use the QR iteration anyway.

Computing, we find that to six decimal places.

$$A^{(10} = \begin{bmatrix} .456399 & -16.7424 & -23.5587 \\ .256970 & 1.54290 & -1.50307 \\ -.000302 & -.000303 & 1.00069 \end{bmatrix} \ .$$

This *almost* has the block structure

$$\begin{bmatrix} .456399 & -16.7424 & * \\ .256970 & 1.54290 & -1.50307 \\ 0 & 0 & 1.00069 \end{bmatrix} .$$

which would mean that the eigenvalues of $A^{(10)}$, and hence of A itself, are 1.00069 and the eigenvalues of $\begin{bmatrix} .456399 & -16.7424 \\ .256970 & 1.54290 \end{bmatrix}$.

The first number is suspiciously close to 1. Is 1 an eigenvalue? Sure enough, the rank of $A-2I$ is 2, and $\begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$ is in the kernel. We have found one eigenvector exactly! Of course, you could also have computed the characteristic polynomial, and found that 1 is eigenvector that way, but using the QR iteration is effective even for very large matrices.

In any case, normalizing the eigenvector $\begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$ that we found, we have $\mathbf{u} = \dfrac{1}{3}\begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$. The Householder reflection matrix M that reflects this vector onto $\mathbf{e}_1$ is

$$M = \frac{1}{3}\begin{bmatrix} 2 & 2 & 1 \\ 2 & -1 & -2 \\ 1 & -2 & 2 \end{bmatrix} .$$

Define U_1 to be M, and compute

$$(U_1)^* A U_1 = MAM = \frac{1}{3}\begin{bmatrix} 1 & -76 & -26 \\ 0 & 13 & 4 \\ 0 & -34 & -7 \end{bmatrix} .$$

We are halfway done! We now take B to be the 2×2 block in the lower right, namely

$$B = \frac{1}{3}\begin{bmatrix} 13 & 4 \\ -34 & -7 \end{bmatrix} .$$

This being a 2×2 matrix, it is a trivial matter to find ists eigenvalues and eigenvectors.

Computing the characteristic polynomial, and using the quadratic formula, we find that the eigenvalues are $1+2i$ and $1-2i$. Computing an eigevector with eigenvalue $1+2i$ and normalizing it, we have

$$\mathbf{u} = \frac{1}{\sqrt{38}}\begin{bmatrix} 2 \\ -5+3i \end{bmatrix} .$$

We could now construct a unitary matrix that has $\mathbf{u}$ as its first column by taking the Householder reflection that reflects $\mathbf{u}$ onto $\mathbf{e}_1$, but in two dimension, things are even easier. We can take the second column to be $\mathbf{u}^\perp$ where

$$\mathbf{u}^\perp = \frac{1}{\sqrt{38}}\begin{bmatrix} 5+3i \\ 2 \end{bmatrix} .$$

(This differes from the real "perp" just be a complex conjugate, to produce orthogonality in C^2). Therefore, we define

$$U_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2/\sqrt{38} & (5+3i)/\sqrt{38} \\ 0 & (-5+3i)/\sqrt{38} & 2/\sqrt{38} \end{bmatrix} .$$

Now compute

$$T = U_2^*(U_1^* A U_1)U_2 = \begin{bmatrix} 1 & (\sqrt{38}/19)(-3+13i) & (\sqrt{38}/57)(-211-81i) \\ 0 & 1+2i & 20/3+10i \\ 0 & 0 & 1-2i \end{bmatrix} ,$$

and

$$Q = U_1U_2 = \begin{bmatrix} 2/3 & \sqrt{38}(-1/114 + i/38) & (\sqrt{38}/19)(2+i) \\ 2/3 & (\sqrt{38}/57)(4-3i) & (1/\sqrt{38})(-3-i) \\ 1/3 & (\sqrt{38}/57)(-7+3i) & (\sqrt{38}/19)(-1-3i) \end{bmatrix} .$$

We have computed the Schur factorization of A! As you can see, this is a job made for a computer, and the answer would look a lot nicer in decimal form. But here it is, exactly. As you can see from T, the eigenvalues of A are 1, $1+2i$ and $1-2i$.

Now let's look at another example. In this one we focus on finding the first eigenvector. This was a bit too easy in Example 1. In the next example, we use the "shifting" procedure introduced at the end of the last section. This can be used not only to make the QR iteration work when two or more eigenvalues have the same magnitude, but also to dramatically speed up the convergence.

Here is how this goes: We run the QR iteration through , say, 10 steps. If the first column looks at all like a multiple of $\mathbf{e}_1$, the top entry in it is a good guess for the largest eigenvalue.

Therefore, let a denote the upper left entry of $A^{(10)}$. We form the new matrix If $(A-aI)$ is not invertible, then a is an exact eigenvalue, and hence any non zero vector in $\mathrm{Ker}(A-aI)$ gives us the eigenvector we seek. If a is not an exact eigenvalue, then $(A-aI)$ is invertible and $(A-aI)^{-1}$ is well defined. The point of the definition is that if $\mathbf{v}$ is an eigenvector of A with eigenvalue μ, then

$$(A-aI)\mathbf{v} = A\mathbf{v} - a\mathbf{v} = (\mu - a)\mathbf{v}$$

and so

$$(A-aI)^{-1}\mathbf{v} = \frac{1}{\mu - a}\mathbf{v} \ .$$

That is, $\mathbf{v}$ is an eigenvector of $(A-aI)^{-1}$ with eigenvalue $(\mu-a)^{-1}$. Conversely, if $\mathbf{v}$ is an eigenvector of $(A-aI)^{-1}$ with eigeenvalue ν, then

$$(A-aI)^{-1}\mathbf{v} = \nu\mathbf{v} \ ,$$

and applying $A-aI$ to both sides, we have

$$\frac{1}{\nu}\mathbf{v} = (A-aI)\mathbf{v} \ ,$$

and hence

$$A\mathbf{v} = \left(a\nu + \frac{1}{\nu}\right)\mathbf{v}$$

which means that $\mathbf{v}$ is also an eigenvector of A. This proves the following theorem:

Theorem 1 (Same eigenvectors) *Let A be any $n \times n$ matrix, and suppose that $A-aI$ is invertible. Then A and $(A-aI)^{-1}$ have the same eigenvectors, so that if $\mathbf{v}$ is an eigenvector of A, then it is also an eigenvector of $(A-aI)^{-1}$ and vice–versa.*

Moreover, if $A\mathbf{v} = \mu\mathbf{v}$, then $(A-a)^{-1}\mathbf{v} = (\mu - a)^{-1}\mathbf{v}$.

Now if a is much closer to some eigenvalue μ_1 of A than it is to any of the others, the largest eigenvalue, by far, of $(A - aI)^{-1}$ will be

$$\frac{1}{\mu_1 - a}$$

since the denominator is almost zero. Recall that the rate of convergence of QR iteration depends on the differences in the size of the eigenvalues, and since we have just produced an eigenvalue that is much larger than all the others – at least if a is a good approximation to μ_1 – the QR iteration will very quickly converge and give us the corresponding eigenvector of $(A-aI)^{-1}$. Very likely, 10 steps of the QR iteration will do the job for $(A-aI)^{-1}$, and give us an eigenvector of $(A - aI)^{-1}$.

By Theorem 3, this is also an eigenvector of A,and so we have found our eigenvector of A without computing hundreds of steps of the iteration!

This method is called the *shifting method*; you can think of subtracting aI from A as a "shift" of A.

Example 2 (Computing a first eigenvector) Let $A = \begin{bmatrix} 2 & 3 & 5 \\ 2 & -3 & 7 \\ 4 & 1 & 1 \end{bmatrix}$.

We run the QR iteration on A. We compute that

$$A^{(10)} = \begin{bmatrix} 7.54234 & * & * \\ -.009351 & * & * \\ .018542 & * & * \end{bmatrix} .$$

We haven't reported the second and third colums because we *do not care* what is in them. We see from what we have shown that

$$A^{(10)} \approx \begin{bmatrix} 7.5 & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix} .$$

In so far as this is true, $\mathbf{e}_1$ is an eigenvector of $A^{(10)}$ with eigenvalue .75. Since A and $A^{(10)}$ are similar, this would mean that .75 is an eigenvalue of A.

Of course, we have rounded things off, so .75 is not exactly an eigenvalue of A. But, it should be pretty close, and so we can use it as the "shifting value" a in our "amplification" procedure to find the corresponding eigenvector $\mathbf{u}$.

Here is how. We first compute $C = (A - .75I)^{-1}$. We don't show the matrix here; it is not very enlightening. The point is that inverse can be computed in closed form without too much effort.

We now compute the QR iteration for C, keeping track of $U^{(k)}$, as defined in (4.7), as well as $C^{(k)}$. By Theorem 1 of Section 1, specifically (4.10), we will then have that

$$CU^{(k)} = U^{(k)}C^{(k)} . \tag{5.1}$$

Doing the computations, for $k = 10$ we find that

$$C^{(10)} = \begin{bmatrix} 21.1941 & * & * \\ -.176300 \times 10^{-23} & * & * \\ .477380 \times 10^{-22} & * & * \end{bmatrix} .$$

This means that to more than 20 decimal places of accuracy, $\mathbf{e}_1$ is an eigenvector of $C^{(10)}$. The eigenvalues is about 21.1941, but we do not really care. The point is that for some λ, we have

$$C^{(10)}\mathbf{e}_1 = \lambda\mathbf{e}_1 \tag{5.2}$$

to extremely high accuracy.

Now go back to (5.1). In so far as (5.2) is exact,

$$CU^{(k)}\mathbf{e}_1 = U^{(k)}C^{(k)}\mathbf{e}_1 = \lambda U^{(k)}\mathbf{e}_1 \ . \tag{5.3}$$

Now let

$$\mathbf{q}_1 = U^{(10)}\mathbf{e}_1 \ . \tag{5.4}$$

Then, with $k = 10$, (5.3) says that

$$C\mathbf{q}_1 = \lambda\mathbf{q}_1 \ .$$

In other words, the first column of $U^{(10)}$ is very, very close to being an eigenvalues of C, and hence, by Theorem 3, of A. The first column of $U^{(10)}$, to 10 digits of accuracy is

$$\mathbf{u} = \begin{bmatrix} .7169179218 \\ .4747659733 \\ .5105153904 \end{bmatrix} \ .$$

This is an eigenvector of A to extremely good accuracy. Indeed, if we compute $A\mathbf{u}$ we find

$$A\mathbf{u} = \begin{bmatrix} 5.410710716 \\ 3.583145657 \\ 3.852953050 \end{bmatrix} \ .$$

The ratio of the first element of $A\mathbf{u}$ to the first element of $\mathbf{u}$ is

$$\frac{5.410710716}{.7169179218} = 7.547182950 \ .$$

If $\mathbf{u}$ is an eigenvector of A, this must be the eigenvalue. Let's see how well it works:

$$\frac{1}{7.547182950}A\mathbf{u} = \begin{bmatrix} .7169179216 \\ .4747659729 \\ .5105153902 \end{bmatrix} \ .$$

This is *extremely* close to $\mathbf{u}$. We have definitely found an eigenvector!

Shifting is not only helpful for speeding up the QR iteration, it is often essential. For example, when the A has complex eigenvalues, and the eigenvalues that are largest in absolute value are a complex conjugate pair $a \pm ib$, then QR iteration will tend to leave a 2×2 block in the upper left corner no matter how many steps you run. Since convergence of the iteration is driven by differences in the sizes of the eigenvalues, and since there is no difference in size at all between the "top two", they don't get disentangled. For this reason you keep seeing a 2×2 block in the upper left of $A^{(k)}$ no matter how large k gets.

Here is what to do: diagonaize the 2×2 block in the upper left of $A^{(k)}$ for, say, $k = 10$. Suppose the eigenvalues of the 2×2 submatrix are $c \pm id$. Pick $c + id$ as the shift value, That is, form the complex matrix $(A - (c + id))^{-1}$, and apply the QR iteration to it. The two eigenvalues $a \pm ib$ of A get changed into the two eigenvalues

$$\frac{1}{(a-c) + i(b-d)} \qquad \text{and} \qquad \frac{1}{(a-c) - i(b+d)}$$

of $(A - (c + id))^{-1}$. In so far as $c \approx a$ and $d \approx b$, the first one is much larger in magnitude than the second. The shift has cured the problem caused by eigenvalues of the same size!

Moreover, even if you *randomly* chose the shift, you would almost surely "break" the equality of the sizes. But let's not be *too* random. A good guess for a good shift is the larger, in absolute value, of the two eigenvalues in the 2×2 block in the upper left of $A^{(k)}$ for, say, $k = 10$. This can be used even if the QR iteration is not getting "hung up". In that case, the 2×2 block in the upper left of $A^{(k)}$ will be approximately upper triangular, and the largest eigenvalue will be approximately equal to the upper left entry. Therefore, choosing the largest eigenvalue in the upper left 2×2 block would give essentially the same shift we chose in Example 2. We will always make this choice. Finally, we can state our algorithm for finding one eigenvector.

Algorithm for finding one eigenvector

We are given an $n \times n$ matrix A, possibly with complex entries. We declare an $n \times n$ matrix variable C, and a complex variable a. and itinitalize them as follows:

$$C \leftarrow A^{(10)} \qquad \text{and} \qquad a \leftarrow 0 \ .$$

That is, as a preparatory step, we compute 10 steps of the QR iteration for A, and use the result to initialize C. The we proceed with the following while loop:

While the first column of C is not a multiple of $\mathbf{e}_1$,

(1) Let ν be the larger in absolute value of the two eigenvalues of the 2×2 submatrix in the upper left of $C^{(10)}$. (If there is a tie, any rule for selecting either one will do). Update

$$a \leftarrow \nu \qquad \text{and}$$

(2) If a is an eigenvalue of A, find any unit vector $\mathbf{u}$ in $\mathrm{Ker}(A - aI)$, exit the while loop, and and return $\mathbf{u}$. Otherwise, form $(A - aI)^{-1}$ and update

$$C \leftarrow (A - aI)^{-1} \ .$$

(3) Run 10 steps of the QR iteration, producing $C^{(10)}$. Update

$$C \leftarrow C^{(10)} \ .$$

After the while loop terminates, normalize the first column of C, and return this as $\mathbf{u}$.

Of course, the while loop would be an infinite loop as written, but if we fix any $\epsilon > 0$, and modify the stopping rule for the while loop to "while the distance from the first column of C to some multiple of $\mathbf{e}_1$ exceeds ϵ", then it will terminate in a finite number of steps. Using this as a subroutine in the Schur factorization algorithm, you could now write a program to compute Schur factorizations in general – and hence to find the eigenvalues of *any* square matrix.

5.2 How things get complex

Now you can see how we can start from a real matrix A, and end up with a complex Schur factorization: We may be required to use a complex shift at some stage. It is here that the complex numbers enter. Here is an example:

Problems

5.1 Let $\mathbf{u} = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. Find a 3×3 unitary matrix U that has $\mathbf{u}$ as its first column. Then find an orthonormal basis for the plane through the origin given by

$$x + y + z = 0 \ .$$

5.2 Let $\mathbf{u} = \frac{1}{\sqrt{6}}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$. Find a 3×3 unitary matrix U that has $\mathbf{u}$ as its first column. Then find an orthonormal basis for the plane through the origin given by

$$x + 2y + z = 0 \ .$$

5.3 Let $A = \begin{bmatrix} -3 & 4 \\ -6 & 7 \end{bmatrix}$. Compute a Schur factorization of A.

5.4 Let $A = \begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix}$. Compute a Schur factorization of A.

5.5 Let $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 2 \\ 3 & 2 & 1 \end{bmatrix}$. Compute a Schur factorization of A.

Section 6: Matrix Exponentials

The Schur factorization has many uses besides revealing eigenvalues. One is computing matrix exponentials. This will be very useful in the next chapter when we study the prediction and description of motion.

If A is any sqaure matrix, it matrix exponential e^A is defined using the power series for the exponential functions:

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k \ .$$

This formula is useful for deriving properties of the exponential function, but not for explicit calculation: It involves an infinite sum, so literal use of it for computation would involve infinitely many steps.

As we shall explaiin here, there is a practical method for explicitly cacluating e^T for any upper triangular matrix T. The Schur factorization allows this method to be applied to any sqaure matrix A.

6.1 The reduction to upper triangular matrices

Let A be any $n \times n$ matrix. We will explain here how to use the Schur decomposition to compute

$$e^{tA} = \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k \ . \tag{6.1}$$

You may have already learned how to do this when A is diagonalizable, but since not every square matrix can be diagonalized, and approach resting on diagonalization has limited applicability. The point about the Schur factorization is that it allows us to reduce the problem to the computation of e^{tT} where T is upper triangular, as we now explain.

The Schur Decomposition Theorem is the basis of a completely general approach. It says that there is a unitary matrix U and an upper triangular matrix T so that

$$A = UTU^* \ .$$

Here is the first important consequence of this:

• *Suppose that $A = UTU^*$ where U is unitary. Then, for all integers $k \geq 1$,*

$$A^k = UT^kU^* \ . \tag{6.2}$$

Indeed,

$$A^2 = (UTU^*)(UTU^*) = UT(U^*U)TU^* = UT^2U^*$$

since $U^*U = I$. This proves (6.2) for $k = 2$. We now proceed by induction: Assume that (6.2) holds for some value of k. Then

$$A^{k+1} = AA^k = (UTU^*)(UT^kU^*) = UT(U^*U)T^kU^* = UT^{k+1}U^* \ .$$

Hence (6.2) is true for $k = 1$ as well.

Next, we use this to show:

- *Suppose that $A = UTU^*$ where U is unitary. Then*

$$e^{tA} = U\left(e^{tT}\right)U^* \ . \tag{6.3}$$

To see this, just combine (6.1) and (6.2):

$$\begin{aligned}
e^{tA} &= \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!} UT^kU^* \\
&= U\left(\sum_{k=0}^{\infty} \frac{t^k}{k!} T^k\right)U^* \\
&= U\left(e^{tT}\right)U^* \ .
\end{aligned}$$

What this tells us is that if we know how to compute the exponential of an upper triangular matrix T, we can compute the exponential of any square matrix A.

6.2 Exponentiating upper triangular matrices

Now for the second step of the program – a method for exponentiating upper triangular matrices.

First, we recall that a matrix T is upper triangular means that

$$T_{i,j} = 0 \qquad \text{whenever} \qquad j > i \ .$$

Now suppose that S and T are two upper triangular $n \times n$ matrices. Then

$$(ST)_{i,j} = (\text{row } i \text{ of } S) \cdot (\text{column } j \text{ of } T) \ .$$

Now every entry of the ith row of S *before* the ith entry is zero, and every entry of the j column of T after the jth entry is zero. Therefore, if $i > j$, then

$$(\text{row } i \text{ of } S) \cdot (\text{column } j \text{ of } T) = 0 \ ,$$

while if $i = j$, the only entry giving a non zero contribution to the dot product is from the ith places, so that

$$(\text{row } i \text{ of } S) \cdot (\text{column } i \text{ of } T) = S_{i,i}T_{i,i} \ .$$

In other words, $(ST)_{i,i} = S_{i,i}T_{i,i}$. This shows the following:

• *Suppose that S and T are two upper triangular $n \times n$ matrices. Then ST is also upper triangular, and for each $i = 1, \ldots, n$,*

$$(ST)_{i,i} = S_{i,i}T_{i,i} \ .$$

In particular, for all integers $k \geq 1$,

$$(T^k)_{i,i} = (T_{i,i})^k \ . \tag{6.4}$$

To deduce (6.4) from what we have shown above, we again use induction. It is trivially true for $k = 1$. Next, supposing it is true for some k, we take $S = T^k$, and then

$$(T^{k+1})_{i,i} = (ST)_{i,i} = S_{i,i}T_{i,i} = (T^k)_{i,i}T_{i,i} = (T_{i,i})^k T_{i,i} = (T_{i,i})^{k+1} \ .$$

This gives us the inductive step, and hence the formula (6.4) holds in general.

To apply this, we observe that for any i with $i \leq i \leq n$,

$$\begin{aligned}
\left(e^T\right)_{i,i} &= \left(\sum_{k=0}^{\infty} \frac{t^k}{k!} T^k\right)_{i,i} \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!} \left(T^k\right)_{i,i} \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!} \left(T^k_{i,i}\right) \\
&= e^{tT_{i,i}}
\end{aligned}$$

Also, for $i > j$,

$$\begin{aligned}
\left(e^T\right)_{i,j} &= \left(\sum_{k=0}^{\infty} \frac{t^k}{k!} T^k\right)_{i,j} \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!} \left(T^k\right)_{i,j} \\
&= \sum_{k=0}^{\infty} 0 = 0
\end{aligned}$$

since each T^k is upper triangular.

This is *almost* enough information to write down e^{tT}. For example, suppose

$$T = \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix} \ . \tag{6.5}$$

Then we know that

$$e^{tT} = \begin{bmatrix} e^t & a \\ 0 & e^{2t} \end{bmatrix} ,$$

for some unknown a. That is, all of the entries on and below the main diagonal are determined – easily – by what we have just deduced.

What about the entries above the diagonal? As long as all of the diagonal entries are distinct, we need just one more simple observation:

• *For any $n \times n$ matrix A, and any number t,*

$$Ae^{tA} = e^{tA}A \ . \tag{6.6}$$

Indeed,

$$\begin{aligned} Ae^{tA} &= A\left(\sum_{k=0}^{\infty} \frac{t^k}{k!}A^k\right) \\ &= \left(\sum_{k=0}^{\infty} \frac{t^k}{k!}A^{k+1}\right) \\ &= \left(\sum_{k=0}^{\infty} \frac{t^k}{k!}A^k\right) A \\ &= e^{tA}A \ . \end{aligned}$$

Applying this with $A = T$ where T is given by (6.5),

$$\begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} e^t & a \\ 0 & e^{2t} \end{bmatrix} = \begin{bmatrix} e^t & a \\ 0 & e^{2t} \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix} \ .$$

Computing both sides, we find

$$\begin{bmatrix} e^t & a + 3e^{2t} \\ 0 & 2e^{2t} \end{bmatrix} = \begin{bmatrix} e^t & 3e^t + 2a \\ 0 & 2e^{2t} \end{bmatrix} \ .$$

Comparing the upper right corners, we see that

$$a + 3e^{2t} = 3e^t + 2a \ .$$

This means that

$$a = 3(e^{2t} - e^t) \ .$$

This gives us the final unknown entry in e^{tT}, and we have

$$e^{tT} = \begin{bmatrix} e^t & 3(e^{2t} - e^t) \\ 0 & e^{2t} \end{bmatrix} \ . \tag{6.7}$$

We can now apply these ideas to obtain a general formula for the 2×2 case. Let

$$T = \begin{bmatrix} \mu_1 & b \\ 0 & \mu_2 \end{bmatrix} . \tag{6.8}$$

The diagonal entries are the eigenvalues of T, and this is why we denote them by μ_1 and μ_2. The upper right entry is simply denoted b since it has no relation to the eigenvalues. We have

$$e^{tT} = \begin{bmatrix} e^{\tau\mu_1} & a \\ 0 & e^{t\mu_2} \end{bmatrix} ,$$

for some unknown a. Again, all of the entries on and below the main diagonal are determined. Next,

$$Te^{tT} = \begin{bmatrix} \mu_1 & b \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} e^{\tau\mu_1} & a \\ 0 & e^{t\mu_2} \end{bmatrix} = \begin{bmatrix} e^{\tau\mu_1} & a\mu_1 + be^{t\mu_2} \\ 0 & e^{t\mu_2} \end{bmatrix} ,$$

and

$$e^{tT}T = \begin{bmatrix} e^{\tau\mu_1} & a \\ 0 & e^{t\mu_2} \end{bmatrix} \begin{bmatrix} \mu_1 & b \\ 0 & \mu_2 \end{bmatrix} = \begin{bmatrix} e^{\tau\mu_1} & be^{t\mu_1} + a\mu_2 \\ 0 & e^{t\mu_2} \end{bmatrix} .$$

Comparing the upper right entries, and using the fact that $Te^{tT} = e^{tT}T$, we see that

$$a\mu_1 + be^{t\mu_2} = be^{t\mu_1} + a\mu_2 \ .$$

Solving for a, we have

$$a = b\frac{e^{t\mu_1} - e^{t\mu_2}}{\mu_1 - \mu_2} ,$$

whenever $\mu_1 \neq \mu_2$. Once again, degenerate eigenvalues cause trouble! But not for long; we will soon take care of these degenerates. In the mean time, as long as $\mu_1 \neq \mu_2$, we have

$$e^{tT} = \begin{bmatrix} e^{\tau\mu_1} & b(e^{t\mu_1} - e^{t\mu_2})/(\mu_1 - \mu_2) \\ 0 & e^{t\mu_2} \end{bmatrix} . \tag{6.9}$$

If you apply this formula to $T = \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix}$, you find (6.7) as before.

Now let's take care of the degenerates. Consider

$$T = \begin{bmatrix} 2 & 3 \\ 0 & 2 \end{bmatrix} .$$

This time we can't apply (6.9) directly; that would require dividing by zero. We therefore introduce

$$T_\epsilon = \begin{bmatrix} 2 + \epsilon & 3 \\ 0 & 2 \end{bmatrix} .$$

As long a $\epsilon \neq 0$, we can apply (6.9) with the results that

$$e^{tT_\epsilon} = \begin{bmatrix} e^{2t+\epsilon t} & 3(e^{2t+\epsilon t} - e^{2t})/\epsilon \\ 0 & e^{2t} \end{bmatrix} . \tag{6.10}$$

Now we take the limit as ϵ tends to zero. Notice that

$$3(e^{2t+\epsilon t} - e^{2t})/\epsilon = 3e^{2t}\left(\frac{e^{\epsilon t} - 1}{\epsilon}\right)$$

and

$$\lim_{\epsilon\to 0}\left(\frac{e^{\epsilon t} - 1}{\epsilon}\right) = \frac{\mathrm{d}}{\mathrm{d}x}e^{tx}\bigg|_{x=0} = t\ .$$

Therefore, we have

$$e^{tT} = \lim_{\epsilon\to 0} e^{tT_\epsilon} = \begin{bmatrix} e^{2t} & 3te^{2t} \\ 0 & e^{2t} \end{bmatrix} . \tag{6.11}$$

Our next example shows how to deal with the 3×3 case. After this, it should be clear how to handle anything. Let

$$T = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 3 \\ 0 & 0 & 3 \end{bmatrix} .$$

We have

$$e^{tT} = \begin{bmatrix} e^t & a & b \\ 0 & e^{2t} & c \\ 0 & 0 & e^{3t} \end{bmatrix} ,$$

for some unknown entries a, b and c. Then

$$Te^{tT} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 3 \\ 0 & 0 & 3 \end{bmatrix}\begin{bmatrix} e^t & a & b \\ 0 & e^{2t} & c \\ 0 & 0 & e^{3t} \end{bmatrix} = \begin{bmatrix} e^t & a + 2e^{2t} & b + 2c + 3e^{3t} \\ 0 & e^{2t} & 2c + 3e^{3t} \\ 0 & 0 & e^{3t} \end{bmatrix}$$

while

$$e^{tT}T = \begin{bmatrix} e^t & a & b \\ 0 & e^{2t} & c \\ 0 & 0 & e^{3t} \end{bmatrix}\begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 3 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} e^t & 2e^{2t} + 2a & 3e^t + 3a + 3b \\ 0 & e^{2t} & 3e^{2t} + 3c \\ 0 & 0 & e^{3t} \end{bmatrix} .$$

Equating entries in the upper right, we have

$$
\begin{aligned}
a + 2e^{2t} &= 2e^t + 2a \\
b + 2c + 3e^{3t} &= 3e^t + 3a + 3b \\
2c + 3e^{3t} &= 3e^{2t} + 3c
\end{aligned}
$$

It is easy to solve the first of these for a, the third for c, and then, with a and c known, we can solve the second equation for b. We obtain

$$
\begin{aligned}
a &= 2(e^{2t} - e^t) \\
b &= (9e^{3t} - 12e^{2t} + 3e^t)/2 \\
c &= 3(e^{3t} - e^{2t}) \ .
\end{aligned}
$$

Therefore,

$$
e^{tT} = \begin{bmatrix} e^t & 2(e^{2t} - e^t) & (9e^{3t} - 12e^{2t} + 3e^t)/2 \\ 0 & e^{2t} & 3(e^{3t} - e^{2t}) \\ 0 & 0 & e^{3t} \end{bmatrix} \ .
$$

We have now covered all of the ideas needed to compute e^{tT} for any upper triangular matrix T. For example, if

$$
T = \begin{bmatrix} 1 & 2 & 3 & 1 & 2 \\ 0 & 2 & 0 & 2 & 3 \\ 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \ ,
$$

we would introduce

$$
T_\epsilon = \begin{bmatrix} 1 & 2 & 3 & 1 & 2 \\ 0 & 2 & 0 & 2 & 3 \\ 0 & 0 & 1+\epsilon & 1 & 2 \\ 0 & 0 & 0 & 2+\epsilon & 2 \\ 0 & 0 & 0 & 0 & 2+2\epsilon \end{bmatrix} \ ,
$$

so that now all of the diagonal entries are distinct. Then we can compute e^{tT_ϵ}, and finally, $\lim_{\epsilon\to 0} e^{tT_\epsilon}$.

6.3 Other matrix functions

What we have done here for evaluating the exponential function at a matrix input can be done for any function $f(t)$ defined in terms of a convergent power series

$$
f(x) = \sum_{k=0}^{\infty} a_k x^k \ .
$$

If, as is the case for $f(x) = e^x$, or $f(x) = \sin(x)$ or $f(x) = \cos(x)$, the radius of convergence of the power series is infinite, then we may replace x by any $n \times n$ matrix A. If however the radius of convergence R is finite, we must restrict consideration to matrices A with $\|A\| < R$ to be sure the power series even converges. But apart from that, nothing really changes when we replace e^x by another function with a convergent power series.

6.4 Relation with the Jordan canonical form

You may be familiar with something called the *Jordan canonical form* of a matrix, and the *Jordan factorization.* If so, you may find this subsection enlightening, and if not you can skip it. The whole point is that the Jordan factorization is *numerically unstable* while the Schur factorization is not. Any practical question that can be answered using the Jordan factorization can be answered using the Schur factorization, and since it can be stably computed, that is the way to go.

Nevertheless, in many texts the Jordan factorization is used to compute matrix exponentials. The Jordan canonical form of an $n \times n$ matrix A is a special upper triangular matrix J in which only the main diagonal and the diagonal just above it have non zero entries. Moreover, the entries above the main diagonal are either 0 or 1. It is a theorem that every matrix A is similar to such a matrix J. That is, there exists an invertible matrix V so that $A = VJV^{-1}$.

Since J has such a simple structure, it is *very* easy to compute e^J, and then, just as with the Schur factorization, this makes it very easy to calculute e^A.

There is a close relation between the Schur factorization and the Jordan factorization. Let $V = QR$ be a QRfactorization of V. Then $A = VJV^{-1}$ becomes

$$A = (QR)J(R^{-1}Q^{-1}) = Q(RJR^{-1})Q^* \ .$$

Let $T = RJR^{-1}$. This is a product of upper triangular matrices, and so it is upper triangular itself. Hence $A = QTQ^*$ is a Schur factorization of A.

However, the Schur factorization has a significant advantage: It is numerically stable, and the Jordan factorization is not.

Excercises

6.1 Compute e^{tT} for $T = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$.

6.2 Compute e^{tT} for $T = \begin{bmatrix} 2 & 2 \\ 0 & 2 \end{bmatrix}$. Verify that $\lim_{t\to 0} e^{tT} = I$ and also that

$$\frac{\mathrm{d}}{\mathrm{d}t} e^{tT} = T e^{tT}$$

by directly computing both sides.

6.3 We can write the matrix T from problem 1.2 as $T = 2I + U$ where $U = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}$. Notice that $U^2 = 0$, and so

$$T^2 = (2I + U)^2 = 4I + 4U + U^2 = 4I + 4U \ .$$

Show that for all integers $k \geq 1$,

$$T^k = 2^k I + k 2^{k-1} U \ .$$

Then use this closed form formula for T^k to evaluate e^{tT} directly from the power series. You should get the same answer that you got in 1.2. This approach works when all of the diagonal entries are the same, so that the diagonal part commutes with the strictly upper diagonal part. It would not work for the next problem, but it does give another way of understanding where the factor of t comes from.

6.4 Compute e^{tT} for $T = \begin{bmatrix} 1 & 2 & 3 & 1 & 2 \\ 0 & 2 & 0 & 2 & 3 \\ 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$.

6.5 **(a)** Compute a Schur factorization $A = UTU^*$ for $A = \begin{bmatrix} 5 & -8 \\ 2 & 5 \end{bmatrix}$.

(b) For the matrix T found in part**(a)**, compute e^T. The eigenvalues of A are complex, so your answer will involve complex exponetials. Use Euler's formula

$$e^{i\theta} = \cos(\theta) + i\sin(\theta)$$

to eliminate the imaginary exponents.

(c) Combine your results for **(a)** and **(b)** to compute e^{tA}.