Linear Algebra and Differential Equations using MATLAB

January 22, 2021

by Martin Golubitsky and Michael Dellnitz

This document was typeset on Friday 22nd January, 2021.

This book is typeset using LaTeX and the STIX and Gillius fonts.

This book uses the **XIMERA** document class.

We will be glad to receive corrections and suggestions for improvement at ximera@math.osu.edu

Contents

Preface

These notes provide an integrated approach to linear algebra and ordinary differential equations based on computers — in this case the software package MATLAB® [1]. We believe that computers can improve the conceptual understanding of mathematics — not just enable the completion of complicated calculations. We use computers in two ways: in linear algebra computers reduce the drudgery of calculations and enable students to focus on concepts and methods, while in differential equations computers display phase portraits graphically and enable students to focus on the qualitative information embodied in solutions rather than just on developing formulas for solutions.

We develop methods for solving both systems of linear equations and systems of (constant coefficient) linear ordinary differential equations. It is generally accepted that linear algebra methods aid in finding closed form solutions to systems of linear differential equations. The fact that the graphical solution of systems of differential equations can motivate concepts (both geometric and algebraic) in linear algebra is less often discussed. These notes begin by solving linear systems of equations (through standard Gaussian elimination theory) and discussing elementary matrix theory. We then introduce simple differential equations — both single equations and planar systems — to motivate the notions of eigenvectors and eigenvalues. In subsequent chapters linear algebra and ODE theory are often mixed.

Regarding differential equations, our purpose is to introduce at the sophomore – junior level ideas from dynamical systems theory. We focus on phase portraits (and time series) rather than on techniques for finding closed form solutions. We assume that now and in the future practicing scientists and mathematicians will use ODE solving computer programs more frequently than they will use techniques of integration. For this reason we have focused on the information that is embedded in the computer graphical approach. We discuss both typical phase portraits (Morse-Smale systems) and typical one parameter bifurcations (both local and global). Our goal is to provide the mathematical background that is needed when interpreting the results of computer simulation.

[1] MATLAB is a registered trademark of The MathWorks Inc. Natick, MA

The integration of computers: Our approach assumes that students have an easier time learning with computers if the computer segments are fully integrated with the course material. So we have interleaved the instructions on how to use MATLAB with the examples and theory in the text. With ease of use in mind, we have also provided a number of preloaded matrices and differential equations with the notes. Any equation label in this text that is followed by an asterisk can be loaded into MATLAB just by typing the formula number. For the successful use of this text, it is important that students have access to computers with MATLAB and the computer files associated with these notes.

John Polking has developed an excellent graphical user interface for solving planar systems of autonomous differential equations called pplane10. We use pplane10 instead of using the MATLAB native commands for solving ODEs. In these notes we also provide an introduction to pplane10 and the other associated software routines.

For the most part we treat the computer as a black box. We have not attempted to explain how the computer, or more precisely MATLAB, performs computations. Linear algebra structures are developed (typically) with proofs, while differential equations theorems are presented (typically) without proof and are instead motivated by computer experimentation.

There are two types of exercises included with most sections — those that should be completed using pencil and paper (called Hand Exercises) and those that should be completed with the assistance of computers (called Computer Exercises).

Ways to use the text: We envision this course as a one-year sequence replacing the standard one semester linear algebra and ODE courses. There is a natural one semester *Linear Systems* course that can be taught using the material in this book. In this course students will learn both the basics of linear algebra and the basics of linear systems of differential equations. This one semester course covers the material in the first eight chapters. The *Linear Systems* course stresses eigenvalues and a baby Jordan normal form theory for 2×2 matrices and culminates in a classification of phase portraits for planar constant coefficient linear systems of differential equations. Time permitting additional linear algebra topics from Chapters 9 and 10 may be included. Such material includes changes of coordinates for linear mappings, and orthogonality including Gram-Schmidt orthonormalization and least squares fitting of data.

We believe that by being exposed to ODE theory a student taking just the first semester of this sequence will gain a better appreciation of linear algebra than will a student who takes a standard one semester introduction to linear algebra. However, a more traditional *Linear Algebra* course can be taught by omitting Chapter 7 and de-emphasizing some of the material in Chapter 6. Then there will be time in a one semester course to cover a selection of the linear algebra topics mentioned at the end of the previous paragraph.

Chapters 1–3 We consider the first two chapters to be introductory material and we attempt to cover this material as quickly as we can. Chapter 1 introduces MATLAB along with elementary remarks on vectors and matrices. In our course we ask the students to read the material in Chapter 1 and to use the computer instructions in that chapter as an entry into MATLAB. In class we cover only the material on dot product. Chapter 2 explains how to solve systems of linear equations and is required for a first course on linear algebra. The proof of the uniqueness of reduced echelon form matrices is not very illuminating for students and can be omitted in classroom discussion. Sections whose material we feel can be omitted are noted by asterisks in the Table of Contents and Section 2.6 is the first example of such a section.

In Chapter 3 we introduce matrix multiplication as a notation that simplifies the presentation of systems of linear equations. We then show how matrix multiplication leads to linear mappings and how linearity leads to the principle of superposition. Multiplication of matrices is introduced as composition of linear mappings, which makes transparent the observation that multiplication of matrices is associative. The chapter ends with a discussion of inverse matrices and the role that inverses play in solving systems of linear equations. The determinant of a 2×2 matrix is introduced and its role in determining matrix inverses is emphasized.

Chapter 4 This chapter provides a nonstandard introduction to differential equations. We begin by emphasizing that solutions to differential equations are functions (or pairs of functions for planar systems). We explain in detail the two ways that we may graph solutions to differential equations (time series and phase space) and how to go back and forth between these two graphical representations. The use of the computer is mandatory in this chapter. Chapter 4 dwells on the qualitative theory of solutions to autonomous ordinary differential equations. In one dimension we discuss the importance of knowing equilibria and their stability so that we can understand the fate of all solutions. In two dimensions we emphasize constant coefficient linear systems and the existence (numerical) of invariant directions (eigendirections). In this way we motivate the introduction of eigenvalues and eigenvectors, which are discussed in detail for 2×2 matrices. Once we know how to compute eigenvalues and eigendirections, we then show how this information coupled with superposition leads to closed form solution to initial value problems, at least when the eigenvalues are real and distinct.

We are not trying to give a thorough grounding in techniques for solving differential equations in Chapter 4; rather we are trying to give an introduction to the ways that modern computer programs will represent graphically solutions to differential equations. We have included, however, a section on separation of variables for those who wish to introduce techniques for finding closed form solutions to single differential equations at this time. Our preference is to omit this section in the *Linear Systems* course as well as to omit the applications in Section 4.2 of the linear growth model in one dimension to interest rates and population dynamics.

Chapter 5 In this chapter we introduce vector space theory: vector spaces, subspaces, spanning sets, linear independence, bases, dimensions and the other basic notions in linear algebra. Since solutions to differential equations naturally reside in function spaces, we are able to illustrate that vector spaces other than $\mathbb{R}^n$ arise naturally. We have found that, depending on time, the proof of the main theorem, which appears in Section 5.6, may be omitted in a first course. The material in these chapters is mandatory in any first course on linear algebra.

Chapter 6 At this juncture the text divides into two tracks: one concerned with the qualitative theory of solutions to linear and nonlinear planar systems of differential equations and one mainly concerned with the development of higher dimensional linear algebra. We begin with a description of the differential equations chapters.

Chapter 6 describes closed form solutions to planar systems of constant coefficient linear differential equations in two different ways: a direct method based on eigenvalues and eigenvectors and a related method based on similarity of matrices. Each method has its virtues and vices. Note that the Jordan normal form theorem for 2×2 matrices is proved when discussing how to solve linear planar systems using similarity of matrices.

Chapters 7, 8, 10, and 11 Chapter 7 discusses determinants, characteristic polynomials, and eigenvalues for $n \times n$ matrices. Chapter 8 presents more advanced material on linear mappings including row rank equals column rank and the matrix representation of mappings in different coordinate systems. The material in Sections 8.1 and 8.2 could be presented directly after Chapter 5, while the material in Section 8.3 explains the geometric meaning of similarity.

Orthogonal bases and orthogonal matrices, least squares and Gram-Schmidt orthonormalization, and symmetric matrices are presented in Chapter 10. This material is very important, but is not required later in the text, and may be omitted.

The Jordan normal form theorem for $n \times n$ matrices is presented in Chapter 11. Diagonalization of matrices with distinct real and complex eigenvalues is presented in the first two sections. The appendices, including the proof of the complete Jordan normal form theorem, are included for completeness and should be omitted in classroom presentations.

The Classroom Use of Computers At the University of Houston we use a classroom with an IBM compatible PC and an overhead display. Lectures are presented three hours a week using a combination of blackboard and computer display. We find it inadvisable to use the computer for more than five minutes at a time; we tend to go back and forth between standard lecture style and computer presentations. (The preloaded matrices and differential equations are important to the smooth use of the computer in class.)

We ask students to enroll in a one hour computer lab where they can practice using the material in the text on a computer, do their homework and additional projects, and ask questions of TA's. Our computer lab happens to have 15 power macs. In addition, we ensure that MATLAB and the laode files are available on student use computers around the campus (which is not always easy). The laode files are on the enclosed CDROM; they may also be downloaded by using a web browser or by anonymous ftp.

Acknowledgements This course was first taught on a pilot basis during the 1995–96 academic year at the University of Houston. We thank the Mathematics Department and the College of Natural Sciences and Mathematics of the University of Houston for providing the resources needed to bring a course such as this to fruition. We gratefully acknowledge John Polking's help in adapting his software for our use and for allowing us access to his code so that we could write companion software for use in linear algebra.

We thank Denny Brown for his advice and his careful readings of the many drafts of this manuscript. We thank Gerhard Dangelmayr, Michael Field, Michael Friedberg, Steven Fuchs, Kimber Gross, Barbara Keyfitz, Charles Peters and David Wagner for their advice on the presentation of the material. We also thank Elizabeth Golubitsky, who has written the companion *Solutions Manual*, for her help in keeping the material accessible and in a proper order. Finally, we thank the students who stayed with this course on an experimental basis and by doing so helped to shape its form.

Houston and Bayreuth — Martin Golubitsky

May, 1998 — Michael Dellnitz

Columbus — Martin Golubitsky

February, 2018 — James Fowler

1 Preliminaries

The subjects of linear algebra and differential equations involve manipulating vector equations. In this chapter we introduce our notation for vectors and matrices — and we introduce MATLAB, a computer program that is designed to perform vector manipulations in a natural way.

We begin, in Section 1.1, by defining vectors and matrices, and by explaining how to add and scalar multiply vectors and matrices. In Section 1.2 we explain how to enter vectors and matrices into MATLAB, and how to perform the operations of addition and scalar multiplication in MATLAB. There are many special types of matrices; these types are introduced in Section 1.3. In the concluding section, we introduce the geometric interpretations of vector addition and scalar multiplication; in addition we discuss the angle between vectors through the use of the dot product of two vectors.

1.1 Vectors and Matrices

In their elementary form, matrices and vectors are just lists of real numbers in different formats. An n-vector is a list of n numbers $(x_1, x_2, \ldots, x_n)$. We may write this vector as a *row* vector as we have just done — or as a *column* vector

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

The set of all (real-valued) n-vectors is denoted by $\mathbb{R}^n$; so points in $\mathbb{R}^n$ are called vectors. The sets $\mathbb{R}^n$ when n is small are very familiar sets. The set $\mathbb{R}^1 = \mathbb{R}$ is the real number line, and the set $\mathbb{R}^2$ is the Cartesian plane. The set $\mathbb{R}^3$ consists of points or vectors in three dimensional space.

An $m \times n$ *matrix* is a rectangular array of numbers with m rows and n columns. A general 2×3 matrix has the form

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}.$$

We use the convention that matrix entries a_{ij} are indexed so that the first subscript i refers to the *row* while the second subscript j refers to the *column*. So the entry a_{21} refers to the matrix entry in the 2^{nd} row, 1^{st} column.

An $n \times m$ matrix A and an $n' \times m'$ matrix B are equal precisely when the sizes of the matrices are equal ($n = n'$ and $m = m'$) and when each of the corresponding entries are equal ($a_{ij} = b_{ij}$).

There is some redundancy in the use of the terms "vector" and "matrix". For example, a row n-vector may be thought of as a $1 \times n$ matrix, and a column n-vector may be thought of as a $n \times 1$ matrix. There are situations where matrix notation is preferable to vector notation and vice-versa.

Addition and Scalar Multiplication of Vectors There are two basic operations on vectors: addition and scalar multiplication. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be n-vectors. Then

$$x + y = (x_1 + y_1, \ldots, x_n + y_n);$$

that is, *vector addition* is defined as componentwise addition.

Similarly, *scalar multiplication* is defined as componentwise multiplication. A *scalar* is just a number. Initially, we use the term scalar to refer to a real number — but later on we sometimes use the term scalar to refer to a *complex* number. Suppose r is a real number; then the multiplication of a vector by the scalar r is defined as

$$rx = (rx_1, \ldots, rx_n).$$

Subtraction of vectors is defined simply as

$$x - y = (x_1 - y_1, \ldots, x_n - y_n).$$

Formally, subtraction of vectors may also be defined as

$$x - y = x + (-1)y.$$

Division of a vector x by a scalar r is defined to be

$$\frac{1}{r}x.$$

The standard difficulties concerning division by zero still hold.

Addition and Scalar Multiplication of Matrices Similarly, we add two $m \times n$ matrices by adding corresponding entries, and we multiply a scalar times a matrix by multiplying each entry of the matrix by that scalar. For example,

$$\begin{pmatrix} 0 & 2 \\ 4 & 6 \end{pmatrix} + \begin{pmatrix} 1 & -3 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 5 & 10 \end{pmatrix}$$

and

$$4\begin{pmatrix} 2 & -4 \\ 3 & 1 \end{pmatrix} = \begin{pmatrix} 8 & -16 \\ 12 & 4 \end{pmatrix}.$$

The main restriction on adding two matrices is that the matrices must be of the same size. So you cannot add a 4×3 matrix to 6×2 matrix — even though they both have twelve entries.

Exercises

1.2 MATLAB

We shall use MATLAB to compute addition and scalar multiplication of vectors in two and three dimensions. This will serve the purpose of introducing some basic MATLAB commands.

Entering Vectors and Vector Operations Begin a MATLAB session. We now discuss how to enter a vector into MATLAB. The syntax is straightforward; to enter the row vector $x = (1, 2, 1)$ type[2]

```
x = [1 2 1]
```

and MATLAB responds with

```
x =
     1     2     1
```

Next we show how easy it is to perform addition and scalar multiplication in MATLAB. Enter the row vector $y = (2, -1, 1)$ by typing

```
y = [2 -1 1]
```

and MATLAB responds with

```
y =
     2    -1     1
```

To add the vectors x and y, type

```
x + y
```

and MATLAB responds with

```
ans =
     3     1     2
```

This vector is easily checked to be the sum of the vectors x and y. Similarly, to perform a scalar multiplication, type

```
2*x
```

which yields

```
ans =
     2     4     2
```

MATLAB subtracts the vector y from the vector x in the natural way. Type

```
x - y
```

to obtain

```
ans =
    -1     3     0
```

We mention two points concerning the operations that we have just performed in MATLAB.

(a) When entering a vector or a number, MATLAB automatically echoes what has been entered. *This echoing can be suppressed by appending a semicolon to the line.* For example, type

```
z = [-1 2 3];
```

and MATLAB responds with a new line awaiting a new command. To see the contents of the vector z just type `z` and MATLAB responds with

[2]MATLAB has several useful line editing features. We point out two here:

(a) Horizontal arrow keys ($\rightarrow, \leftarrow$) move the cursor one space without deleting a character.

(b) Vertical arrow keys ($\uparrow, \downarrow$) recall previous and next command lines.

```
z =
    -1     2     3
```

(b) MATLAB stores in a new vector the information obtained by algebraic manipulation. Type

```
a = 2*x - 3*y + 4*z;
```

Now type `a` to find

```
a =
    -8    15    11
```

We see that MATLAB has created a new row vector a with the correct number of entries.

Note: In order to use the result of a calculation later in a MATLAB session, we need to name the result of that calculation. To recall the calculation `2*x - 3*y + 4*z`, we needed to name that calculation, which we did by typing `a = 2*x - 3*y + 4*z`. Then we were able to recall the result just by typing `a`.

We have seen that we enter a row n vector into MATLAB by surrounding a list of n numbers separated by spaces with square brackets. For example, to enter the 5-vector $w = (1, 3, 5, 7, 9)$ just type

```
w = [1 3 5 7 9]
```

Note that the addition of two vectors is only defined when the vectors have the same number of entries. Trying to add the 3-vector x with the 5-vector w by typing `x + w` in MATLAB yields the warning:

```
??? Error using ==> +
Matrix dimensions must agree.
```

In MATLAB new rows are indicated by typing `;`. For example, to enter the column vector

$$z = \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix},$$

just type:

```
z = [-1; 2; 3]
```

and MATLAB responds with

```
z =
    -1
     2
     3
```

Note that MATLAB will not add a row vector and a column vector. Try typing `x + z`.

Individual entries of a vector can also be addressed. For instance, to display the first component of `z` type `z(1)`.

Entering Matrices Matrices are entered into MATLAB row by row with rows separated either by semicolons or by line returns. To enter the 2×3 matrix

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 1 & 4 & 7 \end{pmatrix},$$

just type

```
A = [2 3 1; 1 4 7]
```

MATLAB has very sophisticated methods for addressing the entries of a matrix. You can directly address individual entries, individual rows, and individual columns. To display the entry in the 1^{st} row, 3^{rd} column of A, type `A(1,3)`. To display the 2^{nd} column of A, type `A(:,2)`; and to display the 1^{st} row of A, type `A(1,:)`. For example, to add the two rows of A and store them in the vector x, just type

```
x = A(1,:) + A(2,:)
```

MATLAB has many operations involving matrices — these will be introduced later, as needed.

Exercises

1.3 Special Kinds of Matrices

There are many matrices that have special forms and hence have special names — which we now list.

- A *square* matrix is a matrix with the same number of rows and columns; that is, a square matrix is an $n \times n$ matrix.

- A *diagonal* matrix is a square matrix whose only nonzero entries are along the main diagonal; that is, $a_{ij} = 0$ if $i \neq j$. The following is a 3×3 diagonal matrix
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}.$$
There is a shorthand in MATLAB for entering diagonal matrices. To enter this 3×3 matrix, type `diag([1 2 3])`.

- The *identity* matrix is the diagonal matrix all of whose diagonal entries equal 1. The $n \times n$ identity matrix is denoted by I_n. This identity matrix is entered in MATLAB by typing `eye(n)`.

- A *zero* matrix is a matrix all of whose entries are 0. A zero matrix is denoted by 0. This notation is ambiguous since there is a zero $m \times n$ matrix for every m and n. Nevertheless, this ambiguity rarely causes any difficulty. In MATLAB, to define an $m \times n$ matrix A whose entries all equal 0, just type `A = zeros(m,n)`. To define an $n \times n$ zero matrix B, type `B = zeros(n)`.

- The *transpose* of an $m \times n$ matrix A is the $n \times m$ matrix obtained from A by interchanging rows and columns. Thus the transpose of the 4×2 matrix
$$\begin{pmatrix} 2 & 1 \\ -1 & 2 \\ 3 & -4 \\ 5 & 7 \end{pmatrix}$$
is the 2×4 matrix
$$\begin{pmatrix} 2 & -1 & 3 & 5 \\ 1 & 2 & -4 & 7 \end{pmatrix}.$$
Suppose that you enter this 4×2 matrix into MATLAB by typing
```
A = [2 1; -1 2; 3 -4; 5 7]
```
The transpose of a matrix A is denoted by A^t. To compute the transpose of A in MATLAB, just type `A'`.

- A *symmetric* matrix is a square matrix whose entries are symmetric about the main diagonal; that is $a_{ij} = a_{ji}$. Note that a symmetric matrix is a square matrix A for which $A^t = A$.

- An *upper triangular* matrix is a square matrix all of whose entries below the main diagonal are 0; that is, $a_{ij} = 0$ if $i > j$. A *strictly upper triangular* matrix is an upper triangular matrix whose diagonal entries are also equal to 0. Similar definitions hold for *lower triangular* and *strictly lower triangular* matrices. The following four 3×3 matrices are examples of upper triangular, strictly upper triangular, lower triangular, and strictly lower triangular matrices:
$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 2 & 4 \\ 0 & 0 & 6 \end{pmatrix} \quad \begin{pmatrix} 0 & 2 & 3 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix}$$
$$\begin{pmatrix} 7 & 0 & 0 \\ 5 & 2 & 0 \\ -4 & 1 & -3 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 \\ 5 & 0 & 0 \\ 10 & 1 & 0 \end{pmatrix}.$$

- A square matrix A is *block diagonal* if
$$A = \begin{pmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_k \end{pmatrix}$$

where each B_j is itself a square matrix. An example of a 5×5 block diagonal matrix with one 2×2 block and one 3×3 block is:

$$\begin{pmatrix} 2 & 3 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 3 & 2 & 4 \\ 0 & 0 & 1 & 1 & 5 \end{pmatrix}.$$

Exercises

1.4 The Geometry of Vector Operations

In this section we discuss the geometry of addition, scalar multiplication, and dot product of vectors. We also use MATLAB graphics to visualize these operations.

Geometry of Addition MATLAB has an excellent graphics language that we shall use at various times to illustrate concepts in both two and three dimensions. In order to make the connections between ideas and graphics more transparent, we will sometimes use previously developed MATLAB programs. We begin with such an example — the illustration of the parallelogram law for vector addition.

Suppose that x and y are two planar vectors. Think of these vectors as line segments from the origin to the points x and y in $\mathbb{R}^2$. We use a program written by T.A. Bryan to visualize $x+y$. In MATLAB type[3]:

```
x = [1 2];
y = [-2 3];
addvec(x,y)
```

The vector x is displayed in blue, the vector y in green, and the vector $x+y$ in red. Note that $x+y$ is just the diagonal of the parallelogram spanned by x and y. A black and white version of this figure is given in Figure 1.

The parallelogram law (the diagonal of the parallelogram spanned by x and y is $x+y$) is equally valid in three dimensions. Use MATLAB to verify this statement by typing:

```
x = [1 0 2];
y = [-1 4 1];
addvec3(x,y)
```

[3]Note that all MATLAB commands are case sensitive — upper and lower case must be correct

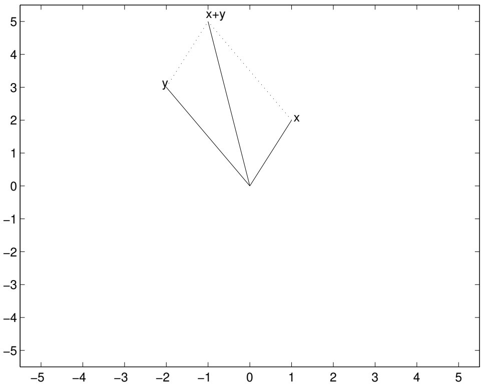

Figure 1: Addition of two planar vectors.

The parallelogram spanned by x and y in $\mathbb{R}^3$ is shown in cyan; the diagonal $x+y$ is shown in blue. See Figure 2. To test your geometric intuition, make several choices of vectors x and y. Note that one vertex of the parallelogram is always the origin.

Geometry of Scalar Multiplication In all dimensions scalar multiplication just scales the length of the vector. To discuss this point we need to define the length of a vector. View an n-vector $x=(x_1,\ldots,x_n)$ as a line segment from the origin to the point x. Using the Pythagorean theorem, it can be shown that the *length* or *norm* of this line segment is:

$$||x|| = \sqrt{x_1^2+\cdots+x_n^2}.$$

MATLAB has the command `norm` for finding the length of a vector. Test this by entering the 3-vector

```
x = [1 4 2];
```

Then type

```
norm(x)
```

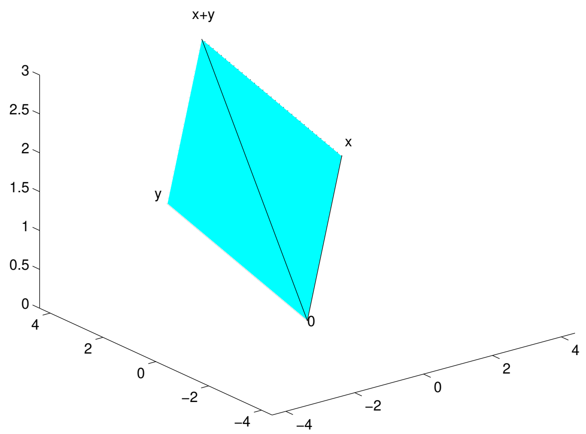

Figure 2: Addition of two vectors in three dimensions.

MATLAB responds with:

```
ans =
    4.5826
```

which is indeed approximately $\sqrt{1+4^2+2^2} = \sqrt{21}$.

Now suppose $r \in \mathbb{R}$ and $x \in \mathbb{R}^n$. A calculation shows that

$$||rx|| = |r|||x||. \qquad (1.4.1)$$

See Exercise **??**. Note also that if r is positive, then the direction of rx is the same as that of x; while if r is negative, then the direction of rx is opposite to the direction of x. The lengths of the vectors $3x$ and $-3x$ are each three times the length of x — but these vectors point in opposite directions. Scalar multiplication by the scalar 0 produces the 0 vector, the vector whose entries are all zero.

Dot Product and Angles The *dot product* of two n-vectors $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ is an important operation on vectors. It is defined by:

$$x \cdot y = x_1 y_1 + \cdots + x_n y_n. \qquad (1.4.2)$$

Note that $x \cdot x$ is just $||x||^2$, the length of x squared.

MATLAB also has a command for computing dot products of n-vectors. Type

```
x = [1 4 2];
y = [2 3 -1];
dot(x,y)
```

MATLAB responds with the dot product of x and y, namely,

```
ans =
    12
```

One of the most important facts concerning dot products is the one that states

$$x \cdot y = 0 \quad \text{if and only if} \quad x \text{ and } y \text{ are perpendicular.} \qquad (1.4.3)$$

Indeed, dot product also gives a way of numerically determining the angle between n-vectors, as follows.

Theorem 1.4.1. *Let θ be the angle between two nonzero n-vectors x and y. Then*

$$\cos\theta = \frac{x \cdot y}{||x||||y||}. \qquad (1.4.4)$$

It follows that $\cos\theta = 0$ if and only if $x \cdot y = 0$. Thus (1.4.3) is valid.

We show that Theorem 1.4.1 is just a restatement of the *law of cosines*. This law states

$$c^2 = a^2 + b^2 - 2ab\cos\theta,$$

where a, b, c are the lengths of the sides of a triangle and θ is the interior angle opposite the side of length c. See Figure 3.

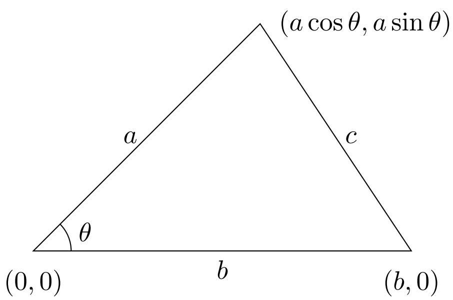

Figure 3: Triangle formed by sides of length a, b, c with interior angle θ opposite side c.

We use trigonometry to verify the law of cosines. First, translate the triangle so that a vertex is at the origin. Second, rotate the triangle placing a vertex on the x-axis and another vertex above the x-axis. After translating and rotating, the coordinates of the nonzero vertex on the x-axis is $(b,0)$. Observe that the vertex above the x-axis has coordinates $(a\cos\theta, a\sin\theta)$. Then use the distance formula to observe that the length c is the distance from the vertex at $(b,0)$ to the vertex at $(a\cos\theta, a\sin\theta)$. That is,

$$\begin{aligned} c^2 &= (a\cos\theta - b)^2 + (a\sin\theta)^2 \\ &= a^2\cos^2\theta - 2ab\cos\theta + b^2 + a^2\sin^2\theta \\ &= a^2 + b^2 - 2ab\cos\theta. \end{aligned}$$

Proof of Theorem 1.4.1 In vector notation we can form a triangle two of whose sides are given by x and y in $\mathbb{R}^n$. The third side is just $x - y$ as $x = y + (x - y)$, as in Figure 4.

It follows from the law of cosines that

$$||x - y||^2 = ||x||^2 + ||y||^2 - 2||x||||y||\cos\theta.$$

We claim that

$$||x - y||^2 = ||x||^2 + ||y||^2 - 2x \cdot y.$$

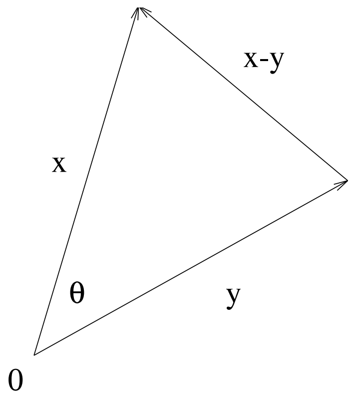

Figure 4: Triangle formed by vectors x and y with interior angle θ.

Assuming that the claim is valid, it follows that

$$x \cdot y = ||x||||y||\cos\theta,$$

which proves the theorem. Finally, compute

$$\begin{aligned} ||x - y||^2 &= (x_1 - y_1)^2 + \cdots + (x_n - y_n)^2 \\ &= (x_1^2 - 2x_1y_1 + y_1^2) + \cdots + (x_n^2 - 2x_ny_n + y_n^2) \\ &= (x_1^2 + \cdots + x_n^2) - 2(x_1y_1 + \cdots + x_ny_n) + (y_1^2 + \cdots \\ &= ||x||^2 - 2x \cdot y + ||y||^2 \end{aligned}$$

to verify the claim. ■

Theorem 1.4.1 gives a numerically efficient method for computing the angle between vectors x and y. In MATLAB this computation proceeds by typing

```
theta = acos(dot(x,y)/(norm(x)*norm(y)))
```

where `acos` is the inverse cosine of a number. For example, using the 3-vectors $x = (1,4,2)$ and $y = (2,3,-1)$ entered previously, MATLAB responds with

```
theta =
    0.7956
```

Remember that this answer is in radians. To convert this answer to degrees, just multiply by 360 and divide by 2π:

```
360*theta / (2*pi)
```

to obtain the answer of $45.5847°$.

Area of Parallelograms Let P be a parallelogram whose sides are the vectors v and w as in Figure 5. Let $|P|$ denote the area of P. As an application of dot products and (1.4.4), we calculate $|P|$. We claim that

$$|P|^2 = ||v||^2||w||^2 - (v \cdot w)^2. \tag{1.4.5}$$

We verify (1.4.5) as follows. Note that the area of P is the same as the area of the rectangle R also pictured in Figure 5. The side lengths of R are: $||v||$ and $||w|| \sin\theta$ where θ is the angle between v and w. A computation using (1.4.4) shows that

$$\begin{aligned} |R|^2 &= ||v||^2||w||^2 \sin^2\theta \\ &= ||v||^2||w||^2(1-\cos^2\theta) \\ &= ||v||^2||w||^2\left(1-\left(\frac{v\cdot w}{||v||||w||}\right)^2\right) \\ &= ||v||^2||w||^2-(v\cdot w)^2, \end{aligned}$$

which establishes (1.4.5).

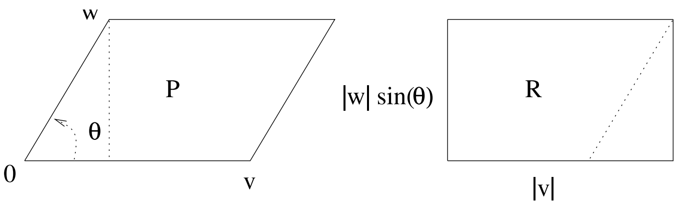

Figure 5: Parallelogram P beside rectangle R with same area.

Exercises

2 Solving Linear Equations

The primary motivation for the study of vectors and matrices is based on the study of solving systems of linear equations. The algorithms that enable us to find solutions are themselves based on certain kinds of matrix manipulations. In these algorithms, matrices serve as a shorthand for calculation, rather than as a basis for a theory. We will see later that these matrix manipulations do lead to a rich theory of how to solve systems of linear equations. But our first step is just to see how these equations are actually solved.

We begin with a discussion in Section 2.1 of how to write systems of linear equations in terms of matrices. We also show by example how complicated writing down the answer to such systems can be. In Section 2.2, we recall that solution sets to systems of linear equations in two and three variables are lines and planes.

The best known and probably the most efficient method for solving systems of linear equations (especially with a moderate to large number of unknowns) is Gaussian elimination. The idea behind this method, which is introduced in Section 2.3, is to manipulate matrices by elementary row operations to reduced echelon form. It is then possible just to look at the reduced echelon form matrix and to read off the solutions to the linear system, if any. The process of reading off the solutions is formalized in Section 2.4; see Theorem 2.4.6. Our discussion of solving linear equations is presented with equations whose coefficients are real numbers — though most of our examples have just integer coefficients. The methods work just as well with complex numbers, and this generalization is discussed in Section 2.5.

Throughout this chapter, we alternately discuss the theory and show how calculations that are tedious when done by hand can easily be performed by computer using MATLAB. The chapter ends with a proof of the uniqueness of row echelon form (a topic of theoretical importance) in Section 2.6. This section is included mainly for completeness and need not be covered on a first reading.

2.1 Systems of Linear Equations and Matrices

It is a simple exercise to solve the system of two equations

$$\begin{array}{rcl} x + y &=& 7 \\ -x + 3y &=& 1 \end{array} \tag{2.1.1}$$

to find that $x = 5$ and $y = 2$. One way to solve system (2.1.1) is to add the two equations, obtaining

$$4y = 8;$$

hence $y = 2$. Substituting $y = 2$ into the 1^{st} equation in (2.1.1) yields $x = 5$.

This system of equations can be solved in a more algorithmic fashion by solving the 1^{st} equation in (2.1.1) for x as

$$x = 7 - y,$$

and substituting this answer into the 2^{nd} equation in (2.1.1), to obtain

$$-(7 - y) + 3y = 1.$$

This equation simplifies to:

$$4y = 8.$$

Now proceed as before.

Solving Larger Systems by Substitution In contrast to solving the simple system of two equations, it is less clear how to solve a complicated system of five equations such as:

$$\begin{array}{rcrcrcrcrcr} 5x_1 &-& 4x_2 &+& 3x_3 &-& 6x_4 &+& 2x_5 &=& 4 \\ 2x_1 &+& x_2 &-& x_3 &-& x_4 &+& x_5 &=& 6 \\ x_1 &+& 2x_2 &+& x_3 &+& x_4 &+& 3x_5 &=& 19 \\ -2x_1 &-& x_2 &-& x_3 &+& x_4 &-& x_5 &=& -12 \\ x_1 &-& 6x_2 &+& x_3 &+& x_4 &+& 4x_5 &=& 4\,. \end{array} \tag{2.1.2}$$

The algorithmic method used to solve (2.1.1) can be expanded to produce a method, called *substitution*, for solving larger systems. We describe the substitution method as it applies to (2.1.2). Solve the 1^{st} equation in (2.1.2) for x_1, obtaining

$$x_1 = \frac{4}{5} + \frac{4}{5}x_2 - \frac{3}{5}x_3 + \frac{6}{5}x_4 - \frac{2}{5}x_5. \tag{2.1.3}$$

Then substitute the right hand side of (2.1.3) for x_1 in the remaining four equations in (2.1.2) to obtain a new system of four equations in the four variables x_2,x_3,x_4,x_5. This procedure eliminates the variable x_1. Now proceed inductively — solve the 1^{st} equation in the new system for x_2 and substitute this expression into the remaining three equations to obtain a system of three equations in three unknowns. This step eliminates the variable x_2. Continue by substitution to eliminate the variables x_3 and x_4, and arrive at a simple equation in x_5 — which can be solved. Once x_5 is known, then x_4, x_3, x_2, and x_1 can be found in turn.

Two Questions

- Is it realistic to expect to complete the substitution procedure without making a mistake in arithmetic?
- Will this procedure work — or will some unforeseen difficulty arise?

Almost surely, attempts to solve (2.1.2) by hand, using the substitution procedure, will lead to arithmetic errors. However, computers and software have developed to the point where solving a system such as (2.1.2) is routine. In this text, we use the software package MATLAB to illustrate just how easy it has become to solve equations such as (2.1.2).

The answer to the second question requires knowledge of the *theory* of linear algebra. In fact, no difficulties will develop when trying to solve the particular system (2.1.2) using the substitution algorithm. We discuss why later.

Solving Equations by MATLAB We begin by discussing the information that is needed by MATLAB to solve (2.1.2). The computer needs to know that there are five equations in five unknowns — but it does not need to keep track of the unknowns $(x_1, x_2, x_3, x_4, x_5)$ by name. Indeed, the computer just needs to know the *matrix of coefficients* in (2.1.2)

$$\begin{pmatrix} 5 & -4 & 3 & -6 & 2 \\ 2 & 1 & -1 & -1 & 1 \\ 1 & 2 & 1 & 1 & 3 \\ -2 & -1 & -1 & 1 & -1 \\ 1 & -6 & 1 & 1 & 4 \end{pmatrix} \tag{2.1.4*}$$

and the *vector* on the right hand side of (2.1.2)

$$\begin{pmatrix} 4 \\ 6 \\ 19 \\ -12 \\ 4 \end{pmatrix}. \tag{2.1.5*}$$

We now describe how we enter this information into MATLAB. To reduce the drudgery and to allow us to focus on ideas, the entries in equations having a $*$ after their label, such as (2.1.4*), have been entered in the `laode` toolbox. This information can be accessed as follows. After starting your MATLAB session, type

```
e2_1_4
```

followed by a carriage return. This instruction tells MATLAB to load equation (2.1.4*) of Chapter 2. The matrix of coefficients is now available in MATLAB; note that this matrix is stored in the 5×5 array `A`. What should appear is:

```
A =
     5    -4     3    -6     2
     2     1    -1    -1     1
     1     2     1     1     3
    -2    -1    -1     1    -1
     1    -6     1     1     4
```

Indeed, comparing this result with (2.1.4*), we see that `A` contains precisely the same information.

Since the label (2.1.5*) is followed by a '$*$', we can enter the vector in (2.1.5*) into MATLAB by typing

```
e2_1_5
```

Note that the right hand side of (2.1.2) is stored in the vector `b`. MATLAB should have responded with

```
b =
     4
     6
    19
   -12
     4
```

Now MATLAB has all the information it needs to solve the system of equations given in (2.1.2). To have MATLAB solve this system, type

```
x = A\b
```

to obtain

```
x =
    5.0000
    2.0000
    3.0000
    4.0000
    1.0000
```

This answer is interpreted as follows: the five values of the unknowns x_1,x_2,x_3,x_4,x_5 are stored in the vector x;

that is,

$$x_1 = 5, \quad x_2 = 2, \quad x_3 = 3, \quad x_4 = 4, \quad x_5 = 1. \quad (2.1.6)$$

The reader may verify that (2.1.6) is indeed a solution of (2.1.2) by substituting the values in (2.1.6) into the equations in (2.1.2).

Changing Entries in MATLAB MATLAB also permits access to single components of x. For instance, type

```
x(5)
```

and the 5^{th} entry of x is displayed,

```
ans =
    1.0000
```

We see that the component `x(i)` of `x` corresponds to the component x_i of the vector x where $i = 1, 2, 3, 4, 5$. Similarly, we can access the entries of the coefficient matrix `A`. For instance, by typing

```
A(3,4)
```

MATLAB responds with

```
ans =
    1
```

It is also possible to change an individual entry in either a vector or a matrix. For example, if we enter

```
A(3,4) = -2
```

we obtain a new matrix `A` which when displayed is:

```
A =
     5    -4     3    -6     2
     2     1    -1    -1     1
     1     2     1    -2     3
    -2    -1    -1     1    -1
     1    -6     1     1     4
```

Thus the command `A(3,4) = -2` changes the entry in the 3^{rd} row, 4^{th} column of `A` from 1 to -2. In other words, we have now entered into MATLAB the information that is needed to solve the system of equations

$$\begin{array}{rcrcrcrcrcr}
5x_1 & - & 4x_2 & + & 3x_3 & - & 6x_4 & + & 2x_5 & = & 4 \\
2x_1 & + & x_2 & - & x_3 & - & x_4 & + & x_5 & = & 6 \\
x_1 & + & 2x_2 & + & x_3 & - & 2x_4 & + & 3x_5 & = & 19 \\
-2x_1 & - & x_2 & - & x_3 & + & x_4 & - & x_5 & = & -12 \\
x_1 & - & 6x_2 & + & x_3 & + & x_4 & + & 4x_5 & = & 4\,.
\end{array}$$

As expected, this change in the coefficient matrix results in a change in the solution of system (2.1.2), as well. Typing

```
x = A\b
```

now leads to the solution

```
x =
    1.9455
    3.0036
    3.0000
    1.7309
    3.8364
```

that is displayed to an accuracy of four decimal places.

In the next step, change `A` as follows:

```
A(2,3) = 1
```

The new system of equations is:

$$\begin{array}{rcrcrcrcrcr}
5x_1 & - & 4x_2 & + & 3x_3 & - & 6x_4 & + & 2x_5 & = & 4\\
2x_1 & + & x_2 & + & x_3 & - & x_4 & + & x_5 & = & 6\\
x_1 & + & 2x_2 & + & x_3 & - & 2x_4 & + & 3x_5 & = & 19\\
-2x_1 & - & x_2 & - & x_3 & + & x_4 & - & x_5 & = & -12\\
x_1 & - & 6x_2 & + & x_3 & + & x_4 & + & 4x_5 & = & 4\,.
\end{array} \tag{2.1.7}$$

The command

```
x = A\b
```

now leads to the message

```
Warning: Matrix is singular to working precision.

x =
   Inf
   Inf
   Inf
   Inf
   Inf
```

Obviously, something is *wrong*; MATLAB cannot find a solution to this system of equations! Assuming that MATLAB is working correctly, we have shed light on one of our previous questions: the method of substitution described by (2.1.3) need *not* always lead to a solution, even though the method does work for system (2.1.2). Why? As we will see, this is one of the questions that is answered by the theory of linear algebra. In the case of (2.1.7), it is fairly easy to see what the difficulty is: the second and fourth equations have the form $y = 6$ and $-y = -12$, respectively.

Warning: The MATLAB command

```
x = A\b
```

may give an error message similar to the previous one. When this happens, one must approach the answer with caution.

Exercises

2.2 The Geometry of Low-Dimensional Solutions

In this section we discuss how to use MATLAB graphics to solve systems of linear equations in two and three unknowns. We begin with two dimensions.

Linear Equations in Two Dimensions The set of all solutions to the equation

$$2x - y = 6 \tag{2.2.1}$$

is a straight line in the xy plane; this line has slope 2 and y-intercept equal to -6. We can use MATLAB to plot the solutions to this equation — though some understanding of the way MATLAB works is needed.

The `plot` command in MATLAB plots a sequence of points in the plane, as follows. Let X and Y be n vectors. Then

```
plot(X,Y)
```

will plot the points $(X(1), Y(1))$, $(X(2), Y(2))$, $\ldots$, $(X(n), Y(n))$ in the xy-plane.

To plot points on the line (2.2.1) we need to enter the x-coordinates of the points we wish to plot. If we want to plot a hundred points, we would be facing a tedious task. MATLAB has a command to simplify this task. Typing

```
x = linspace(-5,5,100);
```

produces a vector x with 100 entries with the 1^{st} entry equal to -5, the last entry equal to 5, and the remaining 98 entries equally spaced between -5 and 5. MATLAB has another command that allows us to create a vector of points `x`. In this command we specify the distance between points rather than the number of points. That command is:

```
x = -5:0.1:5;
```

Producing `x` by either command is acceptable.

Typing

```
y = 2*x - 6;
```

produces a vector whose entries correspond to the y-coordinates of points on the line (2.2.1). Then typing

```
plot(x,y)
```

produces the desired plot. It is useful to label the axes on this figure, which is accomplished by typing

```
xlabel('x')
ylabel('y')
```

We can now use MATLAB to solve the equation (2.1.1) graphically. Recall that (2.1.1) is:

$$\begin{aligned} x + \ \ y &= 7 \\ -x + 3y &= 1 \end{aligned}$$

A solution to this system of equations is a point that lies on both lines in the system. Suppose that we search for a solution to this system that has an x-coordinate between -3 and 7. Then type the commands

```
x = linspace(-3,7,100);
y = 7 - x;
plot(x,y)
xlabel('x')
ylabel('y')
hold on
y = (1 + x)/3;
plot(x,y)
axis('equal')
grid
```

The MATLAB command `hold on` tells MATLAB to keep the present figure and to add the information that follows to that figure. The command `axis('equal')` instructs MATLAB to make unit distances on the x and y axes equal. The last MATLAB command superimposes grid lines. See Figure 6. From this figure you can see that the solution to this system is $(x, y) = (5, 2)$, which we already knew.

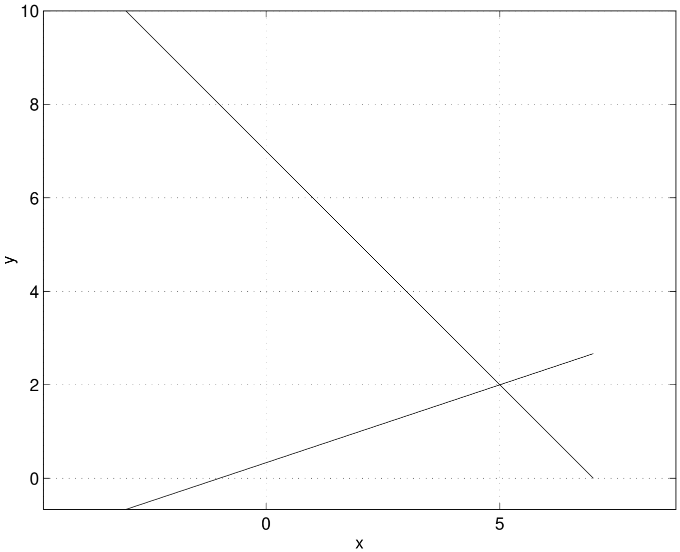

Figure 6: Graph of equations in (2.1.1)

There are several principles that follow from this exercise.

- Solutions to a single linear equation in two variables form a straight line.
- Solutions to two linear equations in two unknowns lie at the intersection of two straight lines in the plane.

It follows that the solution to two linear equations in two variables is a single point if the lines are not parallel. If these lines are parallel and unequal, then there are no solutions, as there are no points of intersection.

Linear Equations in Three Dimensions We begin by observing that the set of all solutions to a linear equation in three variables forms a plane. More precisely, the solutions to the equation

$$ax + by + cz = d \tag{2.2.2}$$

form a plane that is perpendicular to the vector (a, b, c) — assuming of course that the vector (a, b, c) is nonzero.

This fact is most easily proved using the *dot product*. Recall from Chapter 1 (1.4.2) that the dot product is defined by

$$X \cdot Y = x_1 y_1 + x_2 y_2 + x_3 y_3,$$

where $X = (x_1, x_2, x_3)$ and $Y = (y_1, y_2, y_3)$. We recall from Chapter 1 (1.4.3) the following important fact concerning dot products:

$$X \cdot Y = 0$$

if and only if the vectors X and Y are perpendicular.

Suppose that $N = (a, b, c) \neq 0$. Consider the plane that is perpendicular to the *normal vector* N and that contains the point X_0. If the point X lies in that plane, then $X - X_0$ is perpendicular to N; that is,

$$(X - X_0) \cdot N = 0. \tag{2.2.3}$$

If we use the notation

$$X = (x, y, z) \quad \text{and} \quad X_0 = (x_0, y_0, z_0),$$

then (2.2.3) becomes

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0.$$

Setting

$$d = ax_0 + by_0 + cz_0$$

puts equation (2.2.3) into the form (2.2.2). In this way we see that the set of solutions to a single linear equation in three variables forms a plane. See Figure 7.

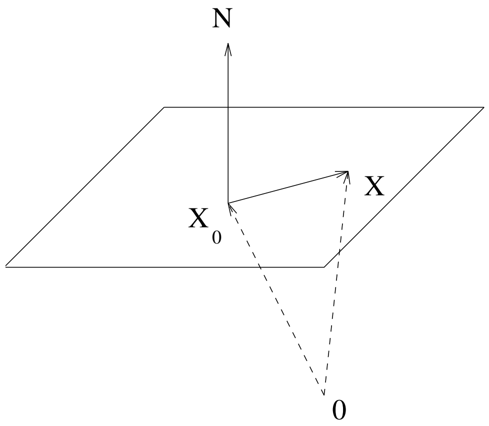

Figure 7: The plane containing X_0 and perpendicular to N.

We now use MATLAB to visualize the planes that are solutions to linear equations. Plotting an equation in three dimensions in MATLAB follows a structure similar to the planar plots. Suppose that we wish to plot the solutions to the equation

$$-2x + 3y + z = 2. \tag{2.2.4}$$

We can rewrite (2.2.4) as

$$z = 2x - 3y + 2.$$

It is this function that we actually graph by typing the commands

```
[x,y] = meshgrid(-5:0.5:5);
z = 2*x - 3*y + 2;
surf(x,y,z)
```

The first command tells MATLAB to create a square grid in the xy-plane. Grid points are equally spaced between -5 and 5 at intervals of 0.5 on both the x and y axes. The second command tells MATLAB to compute the z value of the solution to (2.2.4) at each grid point. The third command tells MATLAB to graph the surface containing the points (x, y, z). See Figure 8.

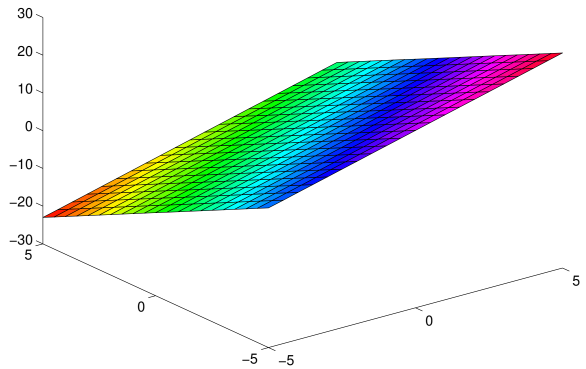

Figure 8: Graph of (2.2.4).

We can now see that solutions to a system of two linear equations in three unknowns consists of points that lie simultaneously on two planes. As long as the normal vectors to these planes are not parallel, the intersection of the two planes will be a line in three dimensions. Indeed, consider the equations

$$\begin{aligned} -2x + 3y + z &= 2 \\ 2x - 3y + z &= 0. \end{aligned}$$

We can graph the solution using MATLAB , as follows. We continue from the previous graph by typing

```
hold on
z = -2*x + 3*y;
surf(x,y,z)
```

The result, which illustrates that the intersection of two planes in $\mathbb{R}^3$ is generally a line, is shown in Figure 9.

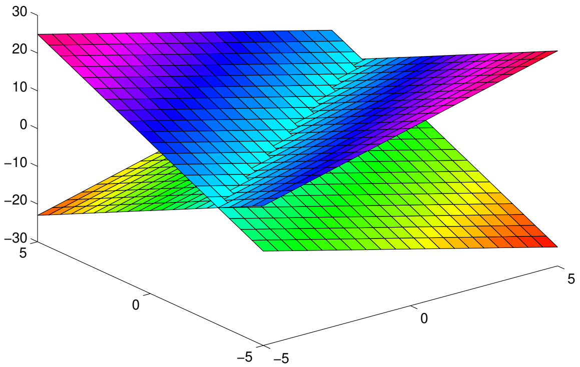

Figure 9: Line of intersection of two planes.

We can now see geometrically that the solution to three simultaneous linear equations in three unknowns will generally be a point — since generally three planes in three space intersect in a point. To visualize this intersection, as shown in Figure 10, we extend the previous system of equations to

$$\begin{aligned} -2x + 3y + z &= 2 \\ 2x - 3y + z &= 0 \\ -3x + 0.2y + z &= 1. \end{aligned}$$

Continuing in MATLAB type

```
z = 3*x - 0.2*y + 1;
surf(x,y,z)
```

Unfortunately, visualizing the point of intersection of these planes geometrically does not really help to get an accurate numerical value of the coordinates of this intersection point. However, we can use MATLAB to solve

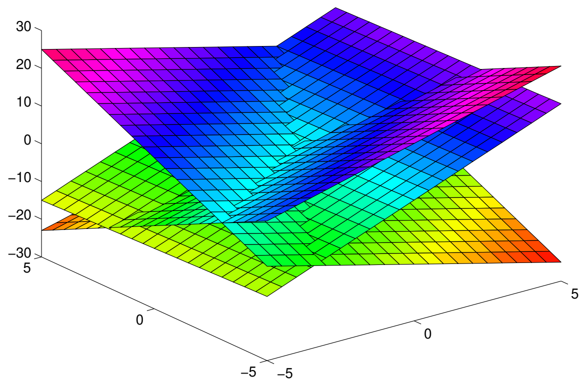

Figure 10: Point of intersection of three planes.

this system accurately. Denote the 3×3 matrix of coefficients by `A`, the vector of coefficients on the right hand side by `b`, and the solution by `x`. Solve the system in MATLAB by typing

```
A = [ -2 3 1; 2 -3 1; -3 0.2 1];
b = [2; 0; 1];
x = A\b
```

The point of intersection of the three planes is at

```
x =
    0.0233
    0.3488
    1.0000
```

Three planes in three dimensional space need not intersect in a single point. For example, if two of the planes are parallel they need not intersect at all. The normal vectors must point in *independent* directions to guarantee that the intersection is a point. Understanding the notion of independence (it is more complicated than just not being parallel) is part of the subject of linear algebra. MATLAB returns "Inf", which we have seen previously, when these normal vectors are (approximately) dependent. For example, consider Exercise ??.

Plotting Nonlinear Functions in MATLAB Suppose that we want to plot the graph of a nonlinear function of a single variable, such as

$$y = x^2 - 2x + 3 \tag{2.2.5}$$

on the interval $[-2, 5]$ using MATLAB. There is a difficulty: How do we enter the term x^2? For example, suppose that we type

```
x = linspace(-2,5);
y = x*x - 2*x + 3;
```

Then MATLAB responds with

```
??? Error using ==> *
Inner matrix dimensions must agree.
```

The problem is that in MATLAB the variable `x` is a vector of 100 equally spaced points `x(1), x(2), ..., x(100)`. What we really need is a vector consisting of entries `x(1)*x(1), x(2)*x(2), ..., x(100)*x(100)`. MATLAB has the facility to perform this operation automatically and the syntax for the operation is `.*` rather than `*`. So typing

```
x = linspace(-2,5);
y = x.*x - 2*x + 3;
plot(x,y)
```

produces the graph of (2.2.5) in Figure 11. In a similar fashion, MATLAB has the 'dot' operations of `./`, `.\`, and `.^`, as well as `.*`.

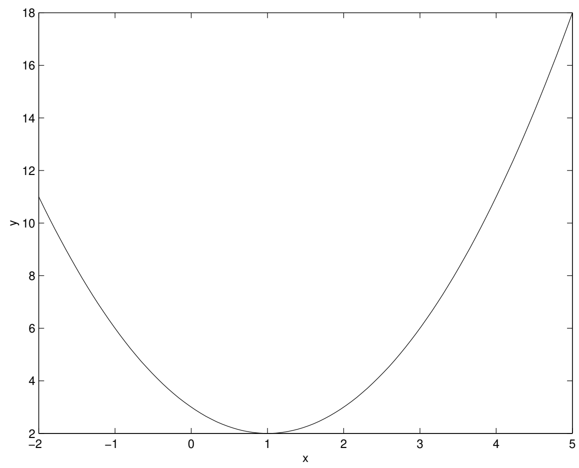

Figure 11: Graph of $y = x^2 - 2x + 3$.

Exercises

2.3 Gaussian Elimination

A general system of m *linear* equations in n unknowns has the form

$$\begin{array}{ccccccc} a_{11}x_1 & + & a_{12}x_2 & + \cdots + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + \cdots + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & \vdots & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + \cdots + & a_{mn}x_n & = & b_m \end{array} \tag{2.3.1}$$

The entries a_{ij} and b_i are constants. Our task is to find a method for solving (2.3.1) for the variables $x_1, \ldots, x_n$.

Easily Solved Equations Some systems are easily solved. The system of three equations ($m = 3$) in three unknowns ($n = 3$)

$$\begin{array}{ccccccc} x_1 & + & 2x_2 & + & 3x_3 & = & 10 \\ & & x_2 & - & \frac{1}{5}x_3 & = & \frac{7}{5} \\ & & & & x_3 & = & 3 \end{array} \tag{2.3.2}$$

is one example. The 3^{rd} equation states that $x_3 = 3$. Substituting this value into the 2^{nd} equation allows us to solve the 2^{nd} equation for $x_2 = 2$. Finally, substituting $x_2 = 2$ and $x_3 = 3$ into the 1^{st} equation allows us to solve for $x_1 = -3$. The process that we have just described is called *back substitution*.

Next, consider the system of two equations ($m = 2$) in three unknowns ($n = 3$):

$$\begin{array}{ccccccc} x_1 & + & 2x_2 & + & 3x_3 & = & 10 \\ & & & & x_3 & = & 3 \end{array} . \tag{2.3.3}$$

The 2^{nd} equation in (2.3.3) states that $x_3 = 3$. Substituting this value into the 1^{st} equation leads to the equation

$$x_1 = 1 - 2x_2.$$

We have shown that every solution to (2.3.3) has the form $(x_1, x_2, x_3) = (1 - 2x_2, x_2, 3)$ and that every vector $(1 - 2x_2, x_2, 3)$ is a solution of (2.3.3). Thus, there is an infinite number of solutions to (2.3.3), and these solutions can be parameterized by one number x_2.

Equations Having No Solutions Note that the system of equations

$$\begin{aligned} x_1 - x_2 &= 1 \\ x_1 - x_2 &= 2 \end{aligned}$$

has no solutions.

Definition 2.3.1. A linear system of equations is *inconsistent* if the system has no solutions and *consistent* if the system does have solutions.

As discussed in the previous section, (2.1.7) is an example of a linear system that MATLAB cannot solve. In fact, that system is inconsistent — inspect the 2^{nd} and 4^{th} equations in (2.1.7).

Gaussian elimination is an algorithm for finding all solutions to a system of linear equations by reducing the given system to ones like (2.3.2) and (2.3.3), that are easily solved by back substitution. Consequently, Gaussian elimination can also be used to determine whether a system is consistent or inconsistent.

Elementary Equation Operations There are three ways to change a system of equations without changing the set of solutions; Gaussian elimination is based on this observation. The three elementary operations are:

(a) Swap two equations.

(b) Multiply a single equation by a nonzero number.

(c) Add a scalar multiple of one equation to another.

We begin with an example:

$$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 10 \\ x_1 & + & 2x_2 & + & x_3 & = & 4 \\ 2x_1 & + & 9x_2 & + & 5x_3 & = & 27 \end{array} \quad . \tag{2.3.4}$$

Gaussian elimination works by eliminating variables from the equations in a fashion similar to the substitution method in the previous section. To begin, eliminate the variable x_1 from all but the 1^{st} equation, as follows. Subtract the 1^{st} equation from the 2^{nd}, and subtract twice the 1^{st} equation from the 3^{rd}, obtaining:

$$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 10 \\ & & & & -2x_3 & = & -6 \\ & & 5x_2 & - & x_3 & = & 7 \end{array} \quad . \tag{2.3.5}$$

Next, swap the 2^{nd} and 3^{rd} equations, so that the coefficient of x_2 in the new 2^{nd} equation is nonzero. This yields

$$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 10 \\ & & 5x_2 & - & x_3 & = & 7 \\ & & & & -2x_3 & = & -6 \end{array} \quad . \tag{2.3.6}$$

Now, divide the 2^{nd} equation by 5 and the 3^{rd} equation by -2 to obtain a system of equations identical to our first example (2.3.2), which we solved by back substitution.

Augmented Matrices The process of performing Gaussian elimination when the number of equations is greater than two or three is painful. The computer, however, can help with the manipulations. We begin by introducing the *augmented matrix*. The augmented matrix associated with (2.3.1) has m rows and $n+1$ columns and is written as:

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array}\right) \tag{2.3.7}$$

The augmented matrix contains all of the information that is needed to solve system (2.3.1).

Elementary Row Operations The elementary operations used in Gaussian elimination can be interpreted as *row operations* on the augmented matrix, as follows:

(a) Swap two rows.

(b) Multiply a single row by a nonzero number.

(c) Add a scalar multiple of one row to another.

We claim that by using these elementary row operations intelligently, we can always solve a consistent linear system — indeed, we can determine when a linear system is consistent or inconsistent. The idea is to perform elementary row operations in such a way that the new augmented matrix has zero entries below the diagonal.

We describe this process inductively. Begin with the 1^{st} column. We assume for now that some entry in this column is nonzero. If $a_{11} = 0$, then swap two rows so that the number a_{11} is nonzero. Then divide the 1^{st} row by a_{11} so that the leading entry in that row is 1. Now subtract a_{i1} times the 1^{st} row from the i^{th} row for each row i from 2 to m. The end result is that the 1^{st} column has a 1 in the 1^{st} row and a 0 in every row below the 1^{st}. The result is

$$\begin{pmatrix} 1 & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots \\ 0 & * & \cdots & * \end{pmatrix}.$$

Next we consider the 2^{nd} column. We assume that some entry in that column below the 1^{st} row is nonzero. So, if necessary, we can swap two rows below the 1^{st} row so that the entry a_{22} is nonzero. Then we divide the 2^{nd} row by a_{22} so that its leading nonzero entry is 1. Then we subtract appropriate multiples of the 2^{nd} row from each row below the 2^{nd} so that all the entries in the 2^{nd} column below the 2^{nd} row are 0. The result is

$$\begin{pmatrix} 1 & * & \cdots & * \\ 0 & 1 & \cdots & * \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & * \end{pmatrix}.$$

Then we continue with the 3^{rd} column. That's the idea. However, does this process always work and what happens if all of the entries in a column are zero? Before answering these questions we do experimentation with MATLAB.

Row Operations in MATLAB In MATLAB the i^{th} row of a matrix `A` is specified by `A(i,:)`. Thus to replace the 5^{th} row of a matrix `A` by twice itself, we need only type:

```
A(5,:) = 2*A(5,:)
```

In general, we can replace the i^{th} row of the matrix `A` by c times itself by typing

```
A(i,:) = c*A(i,:)
```

Similarly, we can divide the i^{th} row of the matrix `A` by the nonzero number c by typing

```
A(i,:) = A(i,:)/c
```

The third elementary row operation is performed similarly. Suppose we want to add c times the i^{th} row to the j^{th} row, then we type

```
A(j,:) = A(j,:) + c*A(i,:)
```

For example, subtracting 3 times the 7^{th} row from the 4^{th} row of the matrix `A` is accomplished by typing:

```
A(4,:) = A(4,:) - 3*A(7,:)
```

The first elementary row operation, swapping two rows, requires a different kind of MATLAB command. In MATLAB, the i^{th} and j^{th} rows of the matrix `A` are permuted by the command

```
A([i j],:) = A([j i],:)
```

So, to swap the 1^{st} and 3^{rd} rows of the matrix `A`, we type

```
A([1 3],:) = A([3 1],:)
```

Examples of Row Reduction in MATLAB Let us see how the row operations can be used in MATLAB. As an example, we consider the augmented matrix

$$\left(\begin{array}{rrrr|r} 1 & 3 & 0 & -1 & -8 \\ 2 & 6 & -4 & 4 & 4 \\ 1 & 0 & -1 & -9 & -35 \\ 0 & 1 & 0 & 3 & 10 \end{array}\right) \tag{2.3.8*}$$

We enter this information into MATLAB by typing

```
e2_3_8
```

which produces the result

```
A =
     1     3     0    -1    -8
     2     6    -4     4     4
     1     0    -1    -9   -35
     0     1     0     3    10
```

We now perform Gaussian elimination on `A`, and then solve the resulting system by back substitution. Gaussian elimination uses elementary row operations to set the entries that are in the lower left part of `A` to zero. These entries are indicated by numbers in the following matrix:

```
*     *     *     *     *
2     *     *     *     *
1     0     *     *     *
0     1     0     *     *
```

Gaussian elimination works inductively. Since the first entry in the matrix A is equal to 1, the first step in Gaussian elimination is to set to zero all entries in the 1^{st} column below the 1^{st} row. We begin by eliminating the `2` that is the first entry in the 2^{nd} row of `A`. We replace the 2^{nd} row by the 2^{nd} row minus twice the 1^{st} row. To accomplish this elementary row operation, we type

```
A(2,:) = A(2,:) - 2*A(1,:)
```

and the result is

```
A =
     1     3     0    -1    -8
     0     0    -4     6    20
     1     0    -1    -9   -35
     0     1     0     3    10
```

In the next step, we eliminate the `1` from the entry in the 3^{rd} row, 1^{st} column of `A`. We do this by typing

```
A(3,:) = A(3,:) - A(1,:)
```

which yields

```
A =
     1     3     0    -1    -8
     0     0    -4     6    20
     0    -3    -1    -8   -27
     0     1     0     3    10
```

Using elementary row operations, we have now set the entries in the 1^{st} column below the 1^{st} row to 0. Next, we alter the 2^{nd} column. We begin by swapping the 2^{nd} and 4^{th} rows so that the leading nonzero entry in the 2^{nd} row is 1. To accomplish this swap, we type

```
A([2 4],:) = A([4 2],:)
```

and obtain

```
A =
     1     3     0    -1    -8
     0     1     0     3    10
     0    -3    -1    -8   -27
     0     0    -4     6    20
```

The next elementary row operation is the command

```
A(3,:) = A(3,:) + 3*A(2,:)
```

which leads to

```
A =
     1     3     0    -1    -8
     0     1     0     3    10
     0     0    -1     1     3
     0     0    -4     6    20
```

Now we have set all entries in the 2^{nd} column below the 2^{nd} row to 0.

Next, we set the first nonzero entry in the 3^{rd} row to `1` by multiplying the 3^{rd} row by -1, obtaining

```
A =
     1     3     0    -1    -8
     0     1     0     3    10
     0     0     1    -1    -3
     0     0    -4     6    20
```

Since the leading nonzero entry in the 3^{rd} row is 1, we next eliminate the nonzero entry in the 3^{rd} column, 4^{th} row. This is accomplished by the following MATLAB command:

```
A(4,:) = A(4,:) + 4*A(3,:)
```

Finally, divide the 4^{th} row by 2 to obtain:

```
A =
     1     3     0    -1    -8
     0     1     0     3    10
     0     0     1    -1    -3
     0     0     0     1     4
```

By using elementary row operations, we have arrived at the system

$$\begin{array}{rcrcrcrcr} x_1 & + & 3x_2 & & & - & x_4 & = & -8 \\ & & x_2 & & & + & 3x_4 & = & 10 \\ & & & & x_3 & - & x_4 & = & -3 \\ & & & & & & x_4 & = & 4\,, \end{array} \tag{2.3.9}$$

that can now be solved by back substitution. We obtain

$$x_4 = 4, \quad x_3 = 1, \quad x_2 = -2, \quad x_1 = 2. \tag{2.3.10}$$

We return to the original set of equations corresponding to (2.3.8*)

$$\begin{array}{rcrcrcrcr} x_1 & + & 3x_2 & & & - & x_4 & = & -8 \\ 2x_1 & + & 6x_2 & - & 4x_3 & + & 4x_4 & = & 4 \\ x_1 & & & - & x_3 & - & 9x_4 & = & -35 \\ & & x_2 & & & + & 3x_4 & = & 10\,. \end{array} \tag{2.3.11*}$$

Load the corresponding linear system into MATLAB by typing

```
e2_3_11
```

The information in (2.3.11*) is contained in the coefficient matrix `C` and the right hand side `b`. A direct solution is found by typing

```
x = C\b
```

which yields the same answer as in (2.3.10), namely,

```
x =
    2.0000
   -2.0000
    1.0000
    4.0000
```

Introduction to Echelon Form Next, we discuss how Gaussian elimination works in an example in which the number of rows and the number of columns in the coefficient matrix are unequal. We consider the augmented matrix

$$\left(\begin{array}{rrrrrr|r} 1 & 0 & -2 & 3 & 4 & 0 & 1 \\ 0 & 1 & 2 & 4 & 0 & -2 & 0 \\ 2 & -1 & -4 & 0 & -2 & 8 & -4 \\ -3 & 0 & 6 & -8 & -12 & 2 & -2 \end{array}\right) \tag{2.3.12*}$$

This information is entered into MATLAB by typing

```
e2_3_12
```

Again, the augmented matrix is denoted by `A`.

We begin by eliminating the `2` in the entry in the 3^{rd} row, 1^{st} column. To accomplish the corresponding elementary row operation, we type

```
A(3,:) = A(3,:) - 2*A(1,:)
```

resulting in

```
A =
     1     0    -2     3     4     0     1
     0     1     2     4     0    -2     0
     0    -1     0    -6   -10     8    -6
    -3     0     6    -8   -12     2    -2
```

We proceed with

```
A(4,:) = A(4,:) + 3*A(1,:)
```

to create two more zeros in the 4^{th} row. Finally, we eliminate the `-1` in the 3^{rd} row, 2^{nd} column by

```
A(3,:) = A(3,:) + A(2,:)
```

to arrive at

```
A =
     1     0    -2     3     4     0     1
     0     1     2     4     0    -2     0
     0     0     2    -2   -10     6    -6
     0     0     0     1     0     2     1
```

Next we set the leading nonzero entry in the 3^{rd} row to 1 by dividing the 3^{rd} row by 2. That is, we type

```
A(3,:) = A(3,:)/2
```

to obtain

```
A =
     1     0    -2     3     4     0     1
     0     1     2     4     0    -2     0
     0     0     1    -1    -5     3    -3
     0     0     0     1     0     2     1
```

We say that the matrix `A` is in (row) *echelon form* since the first nonzero entry in each row is a `1`, each entry in a column below a leading `1` is `0`, and the leading `1` moves to the right as you go down the matrix. In row echelon form, the entries where leading 1's occur are called *pivots*.

If we compare the structure of this matrix to the ones we have obtained previously, then we see that here we have two columns too many. Indeed, we may solve these equations by back substitution for any choice of the variables x_5 and x_6.

The idea behind back substitution is to solve the last equation for the variable corresponding to the first nonzero coefficient. In this case, we use the 4^{th} equation to solve for x_4 in terms of x_5 and x_6, and then we substitute for x_4 in the first three equations. This process can also be accomplished by elementary row operations. Indeed, eliminating the variable x_4 from the first three equations is the same as using row operations to set the first three entries in the 4^{th} column to `0`. We can do this by typing

```
A(3,:) = A(3,:) + A(4,:);
A(2,:) = A(2,:) - 4*A(4,:);
A(1,:) = A(1,:) - 3*A(4,:)
```

Remember: By typing semicolons after the first two rows, we have told MATLAB not to print the intermediate results. Since we have not typed a semicolon after the 3^{rd} row, MATLAB outputs

```
A =
     1     0    -2     0     4    -6    -2
     0     1     2     0     0   -10    -4
     0     0     1     0    -5     5    -2
     0     0     0     1     0     2     1
```

We proceed with back substitution by eliminating the nonzero entries in the first two rows of the 3^{rd} column. To do this, type

```
A(2,:) = A(2,:) - 2*A(3,:);
A(1,:) = A(1,:) + 2*A(3,:)
```

which yields

```
A =
     1     0     0     0    -6     4    -6
     0     1     0     0    10   -20     0
     0     0     1     0    -5     5    -2
     0     0     0     1     0     2     1
```

The augmented matrix is now in *reduced echelon form* and the corresponding system of equations has the form

$$\begin{array}{rrrrrcr} x_1 & & & & - \ 6x_5 & + \ \ 4x_6 & = -6 \\ & x_2 & & & + \ 10x_5 & - \ 20x_6 & = \ \ 0 \\ & & x_3 & & - \ 5x_5 & + \ \ 5x_6 & = -2 \\ & & & x_4 & & + \ \ 2x_6 & = \ \ 1\,, \end{array} \qquad (2.3.13)$$

A matrix is in reduced echelon form if it is in echelon form and if *every* entry in a column containing a pivot, other than the pivot itself, is 0.

Reduced echelon form allows us to solve directly this system of equations in terms of the variables x_5 and x_6,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} -6+6x_5-4x_6 \\ -10x_5+20x_6 \\ -2+5x_5-5x_6 \\ 1-2x_6 \\ x_5 \\ x_6 \end{pmatrix}. \qquad (2.3.14)$$

It is important to note that every consistent system of linear equations corresponding to an augmented matrix in reduced echelon form can be solved as in (2.3.14) — and this is one reason for emphasizing reduced echelon form. We will discuss the reduction to reduced echelon form in more detail in the next section.

Exercises

2.4 Reduction to Echelon Form

In this section, we formalize our previous numerical experiments. We define more precisely the notions of echelon form and reduced echelon form matrices, and we prove that every matrix can be put into reduced echelon form using a sequence of elementary row operations. Consequently, we will have developed an algorithm for determining whether a system of linear equations is consistent or inconsistent, and for determining all solutions to a consistent system.

Definition 2.4.1. A matrix E is in (row) *echelon form* if two conditions hold.

(a) The first nonzero entry in each row of E is equal to 1. This leading entry 1 is called a *pivot*.

(b) A pivot in the $(i+1)^{st}$ row of E occurs in a column to the right of the column where the pivot in the i^{th} row occurs.

Note: A consequence of Definition 2.4.1 is that all rows in an echelon form matrix that are identically zero occur at the bottom of the matrix.

Here are three examples of matrices that are in echelon form. The pivot in each row (which is always equal to 1) is preceded by a $*$.

$$\begin{pmatrix} *1 & 0 & -1 & 0 & -6 & 4 & -6 \\ 0 & *1 & 4 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & *1 & -5 & 5 & -2 \\ 0 & 0 & 0 & 0 & 0 & *1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} *1 & 0 & -1 & 0 & -6 \\ 0 & *1 & 0 & 3 & 0 \\ 0 & 0 & 0 & *1 & -5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & *1 & -1 & 14 & -6 \\ 0 & 0 & 0 & *1 & 15 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Here are three examples of matrices that are *not* in echelon form.

$$\begin{pmatrix} 0 & 0 & 1 & 15 \\ 1 & -1 & 14 & -6 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -1 & 14 & -6 \\ 0 & 0 & 3 & 15 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -1 & 14 & -6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 15 \end{pmatrix}$$

Definition 2.4.2. Two $m \times n$ matrices are *row equivalent* if one can be transformed to the other by a sequence of elementary row operations.

Let $A = (a_{ij})$ be a matrix with m rows and n columns. We want to show that we can perform row operations on A so that the transformed matrix is in echelon form; that is, A is row equivalent to a matrix in echelon form. If $A = 0$, then we are finished. So we assume that some entry in A is nonzero and that the 1^{st} column where that nonzero entry occurs is in the k^{th} column. By swapping rows we can assume that a_{1k} is nonzero. Next, divide the 1^{st} row by a_{1k}, thus setting $a_{1k} = 1$. Now, using MATLAB notation, perform the row operations

```
A(i,:) = A(i,:) - A(i,k)*A(1,:)
```

for each $i \geq 2$. This sequence of row operations leads to a matrix whose first nonzero column has a 1 in the 1^{st} row and a zero in each row below the 1^{st} row.

Now we look for the next column that has a nonzero entry below the 1^{st} row and call that column ℓ. By construction $\ell > k$. We can swap rows so that the entry in the 2^{nd} row, ℓ^{th} column is nonzero. Then we divide the 2^{nd} row by this nonzero element, so that the pivot in the 2^{nd} row is 1. Again we perform elementary row operations so that all entries below the 2^{nd} row in the ℓ^{th} column are set to 0. Now proceed inductively until we run out of nonzero rows.

This argument proves:

Proposition 2.4.3. *Every matrix is row equivalent to a matrix in echelon form.*

More importantly, the previous argument provides an algorithm for transforming matrices into echelon form.

Reduction to Reduced Echelon Form

Definition 2.4.4. A matrix E is in *reduced echelon form* if

(a) E is in echelon form, and

(b) in every column of E having a pivot, every entry in that column other than the pivot is 0.

We can now prove

Theorem 2.4.5. *Every matrix is row equivalent to a matrix in reduced echelon form.*

Proof Let A be a matrix. Proposition 2.4.3 states that we can transform A by elementary row operations to a matrix E in echelon form. Next we transform E into reduced echelon form by some additional elementary row operations, as follows. Choose the pivot in the last nonzero row of E. Call that row ℓ, and let k be the column where the pivot occurs. By adding multiples of the ℓ^{th} row to the rows above, we can transform each entry in the k^{th} column above the pivot to 0. Note that none of these row operations alters the matrix before the k^{th} column. (Also note that this process is identical to the process of back substitution.)

Again we proceed inductively by choosing the pivot in the $(\ell-1)^{st}$ row, which is 1, and zeroing out all entries above that pivot using elementary row operations. ■

Reduced Echelon Form in MATLAB Preprogrammed into MATLAB is a routine to row reduce any matrix to reduced echelon form. The command is `rref`. For example, recall the 4×7 matrix A in (2.3.12*) by typing `e2_3_12`. Put A into reduced row echelon form by typing `rref(A)` and obtaining

```
ans =
     1     0     0     0    -6     4    -6
     0     1     0     0    10   -20     0
     0     0     1     0    -5     5    -2
     0     0     0     1     0     2     1
```

Compare the result with the system of equations (2.3.13).

Solutions to Systems of Linear Equations Originally, we introduced elementary row operations as operations that do not change solutions to the linear system. More precisely, we discussed how solutions to the original system are still solutions to the transformed system and how no new solutions are introduced by elementary row operations. This argument is most easily seen by observing that

all elementary row operations are invertible

— they can be undone.

For example, swapping two rows is undone by just swapping these rows again. Similarly, multiplying a row by a nonzero number c is undone by just dividing that same row by c. Finally, adding c times the j^{th} row to the i^{th} row is undone by subtracting c times the j^{th} row from the i^{th} row.

Thus, we can make several observations about solutions to linear systems. Let E be an augmented matrix corresponding to a system of linear equations having n variables. Since an augmented matrix is formed from the matrix of coefficients by adding a column, we see that the augmented matrix has $n+1$ columns.

Theorem 2.4.6. *Suppose that E is an $m \times (n+1)$ augmented matrix that is in reduced echelon form. Let ℓ be the number of nonzero rows in E*

(a) The system of linear equations corresponding to E is inconsistent if and only if the ℓ^{th} row in E has a pivot in the $(n+1)^{st}$ column.

(b) If the linear system corresponding to E is consistent, then the set of all solutions is parameterized by $n-\ell$ parameters.

Proof Suppose that the last nonzero row in E has its pivot in the $(n+1)^{st}$ column. Then the corresponding equation is:

$$0x_1 + 0x_2 + \cdots + 0x_n = 1,$$

which has no solutions. Thus the system is inconsistent.

Conversely, suppose that the last nonzero row has its pivot before the last column. Without loss of generality, we can renumber the columns — that is, we can renumber the variables x_j — so that the pivot in the i^{th} row occurs in the i^{th} column, where $1 \leq i \leq \ell$. Then the associated system of linear equations has the form:

$$\begin{array}{rcl} x_1 + a_{1,\ell+1}x_{\ell+1} + \cdots + a_{1,n}x_n & = & b_1 \\ x_2 + a_{2,\ell+1}x_{\ell+1} + \cdots + a_{2,n}x_n & = & b_2 \\ \vdots & & \vdots \\ x_\ell + a_{\ell,\ell+1}x_{\ell+1} + \cdots + a_{\ell,n}x_n & = & b_\ell . \end{array}$$

This system can be rewritten in the form:

$$\begin{array}{rcl} x_1 & = & b_1 - a_{1,\ell+1}x_{\ell+1} - \cdots - a_{1,n}x_n \\ x_2 & = & b_2 - a_{2,\ell+1}x_{\ell+1} - \cdots - a_{2,n}x_n \\ \vdots & & \vdots \\ x_\ell & = & b_\ell - a_{\ell,\ell+1}x_{\ell+1} - \cdots - a_{\ell,n}x_n . \end{array} \tag{2.4.1}$$

Thus, each choice of the $n-\ell$ numbers $x_{\ell+1}, \ldots, x_n$ uniquely determines values of $x_1, \ldots, x_\ell$ so that $x_1, \ldots, x_n$ is a solution to this system. In particular, the system is consistent, so (a) is proved; and the set of all solutions is parameterized by $n-\ell$ numbers, so (b) is proved. ■

Two Examples Illustrating Theorem 2.4.6 The reduced echelon form matrix

$$E = \left(\begin{array}{ccc|c} 1 & 5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

is the augmented matrix of an inconsistent system of three equations in three unknowns.

The reduced echelon form matrix

$$E = \left(\begin{array}{ccc|c} 1 & 5 & 0 & 2 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

is the augmented matrix of a consistent system of three equations in three unknowns x_1, x_2, x_3. For this matrix $n=3$ and $\ell=2$. It follows from Theorem 2.4.6 that the solutions to this system are specified by one parameter. Indeed, the solutions are

$$\begin{array}{rcl} x_1 & = & 2 - 5x_2 \\ x_3 & = & 5 \end{array}$$

and are specified by the one parameter x_2.

Consequences of Theorem 2.4.6 It follows from Theorem 2.4.6 that linear systems of equations with fewer equations than unknowns and with zeros on the right hand side always have nonzero solutions. More precisely:

Corollary 2.4.7. *Let A be an $m \times n$ matrix where $m < n$. Then the system of linear equations whose augmented matrix is $(A|0)$ has a nonzero solution.*

Proof Perform elementary row operations on the augmented matrix $(A|0)$ to arrive at the reduced echelon form matrix $(E|0)$. Since the zero vector is a solution, the associated system of equations is consistent. Now the number of nonzero rows ℓ in $(E|0)$ is less than or equal to the number of rows m in E. By assumption $m < n$ and hence $\ell < n$. It follows from Theorem 2.4.6 that solutions to the linear system are parametrized by $n - \ell \geq 1$ parameters and that there are nonzero solutions. ∎

Recall that two $m \times n$ matrices are row equivalent if one can be transformed to the other by elementary row operations.

Corollary 2.4.8. *Let A be an $n \times n$ square matrix and let b be in $\mathbb{R}^n$. Then A is row equivalent to the identity matrix I_n if and only if the system of linear equations whose augmented matrix is $(A|b)$ has a unique solution.*

Proof Suppose that A is row equivalent to I_n. Then, by using the same sequence of elementary row operations, it follows that the $n \times (n+1)$ augmented matrix $(A|b)$ is row equivalent to $(I_n|c)$ for some vector $c \in \mathbb{R}^n$. The system of linear equations that corresponds to $(I_n|c)$ is:

$$\begin{array}{ccc} x_1 & = & c_1 \\ \vdots & \vdots & \vdots \\ x_n & = & c_n, \end{array}$$

which transparently has the unique solution $x = (c_1, \ldots, c_n)$. Since elementary row operations do not change the solutions of the equations, the original augmented system $(A|b)$ also has a unique solution.

Conversely, suppose that the system of linear equations associated to $(A|b)$ has a unique solution. Suppose that $(A|b)$ is row equivalent to a reduced echelon form matrix E. Suppose that the last nonzero row in E is the ℓ^{th} row. Since the system has a solution, it is consistent. Hence Theorem 2.4.6(b) implies that the solutions to the system corresponding to E are parameterized by $n - \ell$ parameters. If $\ell < n$, then the solution is not unique. So $\ell = n$.

Next observe that since the system of linear equations is consistent, it follows from Theorem 2.4.6(a) that the pivot in the n^{th} row must occur in a column before the $(n+1)^{st}$. It follows that the reduced echelon matrix $E = (I_n|c)$ for some $c \in \mathbb{R}^n$. Since $(A|b)$ is row equivalent to $(I_n|c)$, it follows, by using the same sequence of elementary row operations, that A is row equivalent to I_n. ∎

Uniqueness of Reduced Echelon Form and Rank Abstractly, our discussion of reduced echelon form has one point remaining to be proved. We know that every matrix A can be transformed by elementary row operations to reduced echelon form. Suppose, however, that we use two different sequences of elementary row operations to transform A to two reduced echelon form matrices E_1 and E_2. Can E_1 and E_2 be different? The answer is: No.

Theorem 2.4.9. *For each matrix A, there is precisely one reduced echelon form matrix E that is row equivalent to A.*

The proof of Theorem 2.4.9 is given in Section 2.6. Since every matrix is row equivalent to a unique matrix in reduced echelon form, we can define the rank of a matrix as follows.

Definition 2.4.10. Let A be an $m \times n$ matrix that is row equivalent to a reduced echelon form matrix E. Then the *rank* of A, denoted $\text{rank}(A)$, is the number of nonzero rows in E.

We make three remarks concerning the rank of a matrix.

- An echelon form matrix is always row equivalent to a reduced echelon form matrix with the same number of nonzero rows. Thus, to compute the rank of a matrix, we need only perform elementary row operations until the matrix is in echelon form.
- The rank of any matrix is easily computed in MATLAB. Enter a matrix A and type `rank(A)`.
- The number ℓ in the statement of Theorem 2.4.6 is just the rank of E.

In particular, if the rank of the augmented matrix corresponding to a consistent system of linear equations in n unknowns has rank ℓ, then the solutions to this system are parametrized by $n - \ell$ parameters.

Exercises

2.5 Linear Equations with Special Coefficients

In this chapter we have shown how to use elementary row operations to solve systems of linear equations. We have assumed that each linear equation in the system has the form

$$a_{j1}x_1 + \cdots + a_{jn}x_n = b_j,$$

where the a_{ji}s and the b_js are real numbers. For simplicity, in our examples we have only chosen equations with integer coefficients — such as:

$$2x_1 - 3x_2 + 15x_3 = -1.$$

Systems with Nonrational Coefficients In fact, a more general choice of coefficients for a system of two equations might have been

$$\begin{array}{rcl} \sqrt{2}x_1 + 2\pi x_2 & = & 22.4 \\ 3x_1 + 36.2x_2 & = & e. \end{array} \tag{2.5.1}$$

Suppose that we solve (2.5.1) by elementary row operations. In matrix form we have the augmented matrix

$$\left(\begin{array}{cc|c} \sqrt{2} & 2\pi & 22.4 \\ 3 & 36.2 & e \end{array}\right).$$

Proceed with the following elementary row operations. Divide the 1^{st} row by $\sqrt{2}$ to obtain

$$\left(\begin{array}{cc|c} 1 & \pi\sqrt{2} & 11.2\sqrt{2} \\ 3 & 36.2 & e \end{array}\right).$$

Next, subtract 3 times the 1^{st} row from the 2^{nd} row to obtain:

$$\left(\begin{array}{cc|c} 1 & \pi\sqrt{2} & 11.2\sqrt{2} \\ 0 & 36.2 - 3\pi\sqrt{2} & e - 33.6\sqrt{2} \end{array}\right).$$

Then divide the 2^{nd} row by $36.2 - 3\pi\sqrt{2}$, obtaining:

$$\left(\begin{array}{cc|c} 1 & \pi\sqrt{2} & 11.2\sqrt{2} \\ 0 & 1 & \dfrac{e - 33.6\sqrt{2}}{36.2 - 3\pi\sqrt{2}} \end{array}\right).$$

Finally, multiply the 2^{nd} row by $\pi\sqrt{2}$ and subtract it from the 1^{st} row to obtain:

$$\left(\begin{array}{cc|c} 1 & 0 & 11.2\sqrt{2} - \pi\sqrt{2}\dfrac{e - 33.6\sqrt{2}}{36.2 - 3\pi\sqrt{2}} \\ 0 & 1 & \dfrac{e - 33.6\sqrt{2}}{36.2 - 3\pi\sqrt{2}} \end{array}\right).$$

So

$$\begin{array}{rcl} x_1 & = & 11.2\sqrt{2} - \pi\sqrt{2}\dfrac{e - 33.6\sqrt{2}}{36.2 - 3\pi\sqrt{2}} \\ x_2 & = & \dfrac{e - 33.6\sqrt{2}}{36.2 - 3\pi\sqrt{2}} \end{array} \tag{2.5.2}$$

which is both hideous to look at and quite uninformative. It is, however, correct.

Both x_1 and x_2 are real numbers — they had to be because all of the manipulations involved addition, subtraction, multiplication, and division of real numbers — which yield real numbers.

If we wanted to use MATLAB to perform these calculations, we have to convert $\sqrt{2}$, π, and e to their decimal equivalents — at least up to a certain decimal place accuracy. This introduces errors — which for the moment we assume are small.

To enter A and b in MATLAB , type

```
A = [sqrt(2) 2*pi; 3 36.2];
b = [22.4; exp(1)];
```

Now type `A` to obtain:

```
A =
    1.4142    6.2832
    3.0000   36.2000
```

As its default display, MATLAB displays real numbers to four decimal place accuracy. Similarly, type `b` to obtain

```
b =
   22.4000
    2.7183
```

Next use MATLAB to solve this system by typing:

```
A\b
```

to obtain

```
ans =
   24.5417
   -1.9588
```

The reader may check that this answer agrees with the answer in (2.5.2) to MATLAB output accuracy by typing

```
x2 = (exp(1)-33.6*sqrt(2))/(36.2-3*pi*sqrt(2))
x1 = 11.2*sqrt(2)-pi*sqrt(2)*x2
```

to obtain

```
x1 =
   24.5417
```

and

```
x2 =
   -1.9588
```

More Accuracy MATLAB can display numbers in machine precision (15 digits) rather than the standard four decimal place accuracy. To change to this display, type

```
format long
```

Now solve the system of equations (2.5.1) again by typing

```
A\b
```

and obtaining

```
ans =
  24.54169560069650
  -1.95875151860858
```

Integers and Rational Numbers Now suppose that all of the coefficients in a system of linear equations are integers. When we add, subtract or multiply integers — we get integers. In general, however, when we divide an integer by an integer we get a rational number rather than an integer. Indeed, since elementary row operations involve only the operations of addition, subtraction, multiplication and division, we see that if we perform elementary row operations on a matrix with integer entries, we will end up with a matrix with rational numbers as entries.

MATLAB can display calculations using rational numbers rather than decimal numbers. To display calculations using only rational numbers, type

```
format rational
```

For example, let

$$A = \begin{pmatrix} 2 & 2 & 1 & 0 \\ 1 & 3 & -5 & 1 \\ 4 & 2 & 1 & 3 \\ 2 & 1 & -1 & 4 \end{pmatrix} \tag{2.5.3*}$$

and let

$$b = \begin{pmatrix} 1 \\ 1 \\ -5 \\ 2 \end{pmatrix}. \tag{2.5.4*}$$

Enter A and b into MATLAB by typing

```
e2_5_3
e2_5_4
```

Solve the system by typing

```
A\b
```

to obtain

```
ans =
  -357/41
   309/41
   137/41
   156/41
```

To display the answer in standard decimal form, type

```
format
A\b
```

obtaining

```
ans =
   -8.7073
    7.5366
    3.3415
    3.8049
```

The same logic shows that if we begin with a system of equations whose coefficients are rational numbers, we will obtain an answer consisting of rational numbers — since adding, subtracting, multiplying and dividing rational numbers yields rational numbers. More precisely:

Theorem 2.5.1. *Let A be an $n \times n$ matrix that is row equivalent to I_n, and let b be an n vector. Suppose that all entries of A and b are rational numbers. Then there is a unique solution to the system corresponding to the augmented matrix $(A|b)$ and this solution has rational numbers as entries.*

Proof Since A is row equivalent to I_n, Corollary 2.4.8 states that this linear system has a unique solution x. As we have just discussed, solutions are found using elementary row operations — hence the entries of x are rational numbers. ∎

Complex Numbers In the previous parts of this section, we have discussed why solutions to linear systems whose coefficients are rational numbers must themselves have entries that are rational numbers. We now discuss solving linear equations whose coefficients are more general than real numbers; that is, whose coefficients are complex numbers.

First recall that addition, subtraction, multiplication and division of complex numbers yields complex numbers. Suppose that

$$\begin{aligned} a &= \alpha + i\beta \\ b &= \gamma + i\delta \end{aligned}$$

where $\alpha, \beta, \gamma, \delta$ are real numbers and $i = \sqrt{-1}$. Then

$$\begin{aligned} a + b &= (\alpha + \gamma) + i(\beta + \delta) \\ a - b &= (\alpha - \gamma) + i(\beta - \delta) \\ ab &= (\alpha\gamma - \beta\delta) + i(\alpha\delta + \beta\gamma) \\ \frac{a}{b} &= \frac{\alpha\gamma + \beta\delta}{\gamma^2 + \delta^2} + i\frac{\beta\gamma - \alpha\delta}{\gamma^2 + \delta^2} \end{aligned}$$

MATLAB has been programmed to do arithmetic with complex numbers using exactly the same instructions as it uses to do arithmetic with real and rational numbers. For example, we can solve the system of linear equations

$$\begin{array}{rcl} (4-i)x_1 + 2x_2 & = & 3-i \\ 2x_1 + (4-3i)x_2 & = & 2+i \end{array}$$

in MATLAB by typing

```
A = [4-i 2; 2 4-3i];
b = [3-i; 2+i];
A\b
```

The solution to this system of equations is:

```
ans =
   0.8457 - 0.1632i
  -0.1098 + 0.2493i
```

Note: Care must be given when entering complex numbers into arrays in MATLAB. For example, if you type

```
b = [3 -i; 2 +i]
```

then MATLAB will respond with the 2×2 matrix

```
b =
   3.0000                   0 - 1.0000i
   2.0000                   0 + 1.0000i
```

Typing either `b = [3-i; 2+i]` or `b = [3 - i; 2 + i]` will yield the desired 2×1 column vector.

All of the theorems concerning the existence and uniqueness of row echelon form — and for solving systems of linear equations — work when the coefficients of the linear system are complex numbers as opposed to real numbers. In particular:

Theorem 2.5.2. *If the coefficients of a system of n linear equations in n unknowns are complex numbers and if the coefficient matrix is row equivalent to I_n, then there is a unique solution to this system whose entries are complex numbers.*

Complex Conjugation Let $a = \alpha + i\beta$ be a complex number. Then the *complex conjugate* of a is defined to be

$$\overline{a} = \alpha - i\beta.$$

Let $a = \alpha + i\beta$ and $c = \gamma + i\delta$ be complex numbers. Then we claim that

$$\begin{array}{rcl} \overline{a+c} & = & \overline{a} + \overline{c} \\ \overline{ac} & = & \overline{a}\,\overline{c} \end{array} \tag{2.5.5}$$

To verify these statements, calculate

$$\begin{aligned} \overline{a+c} &= \overline{(\alpha+\gamma) + i(\beta+\delta)} = (\alpha+\gamma) - i(\beta+\delta) \\ &= (\alpha - i\beta) + (\gamma - i\delta) = \overline{a} + \overline{c} \end{aligned}$$

and

$$\begin{aligned} \overline{ac} &= \overline{(\alpha\gamma - \beta\delta) + i(\alpha\delta + \beta\gamma)} \\ &= (\alpha\gamma - \beta\delta) - i(\alpha\delta + \beta\gamma) \\ &= (\alpha - i\beta)(\gamma - i\delta) = \overline{a}\,\overline{c}. \end{aligned}$$

Exercises

2.6 Uniqueness of Reduced Echelon Form

In this section we prove Theorem 2.4.9, which states that every matrix is row equivalent to precisely one reduced echelon form matrix.

Proof of Theorem 2.4.9: Suppose that E and F are two $m \times n$ reduced echelon matrices that are row equivalent to A. Since elementary row operations are invertible, the two matrices E and F are row equivalent. Thus, the systems of linear equations associated to the $m \times (n+1)$ matrices $(E|0)$ and $(F|0)$ must have exactly the same set of solutions. It is the fact that the solution sets of the linear equations associated to $(E|0)$ and $(F|0)$ are identical that allows us to prove that $E = F$.

Begin by renumbering the variables $x_1, \ldots, x_n$ so that the equations associated to $(E|0)$ have the form:

$$\begin{array}{rcl} x_1 & = & -a_{1,\ell+1}x_{\ell+1} - \cdots - a_{1,n}x_n \\ x_2 & = & -a_{2,\ell+1}x_{\ell+1} - \cdots - a_{2,n}x_n \\ \vdots & & \vdots \\ x_\ell & = & -a_{\ell,\ell+1}x_{\ell+1} - \cdots - a_{\ell,n}x_n. \end{array} \tag{2.6.1}$$

In this form, pivots of E occur in the columns $1, \ldots, \ell$. We begin by showing that the matrix F also has pivots in columns $1, \ldots, \ell$. Moreover, there is a unique solution to these equations for *every* choice of numbers $x_{\ell+1}, \ldots, x_n$.

Suppose that the pivots of F do not occur in columns $1, \ldots, \ell$. Then there is a row in F whose first nonzero entry occurs in a column $k > \ell$. This row corresponds to an equation

$$x_k = c_{k+1}x_{k+1} + \cdots + c_n x_n.$$

Now, consider solutions that satisfy

$$x_{\ell+1} = \cdots = x_{k-1} = 0 \quad \text{and} \quad x_{k+1} = \cdots = x_n = 0.$$

In the equations associated to the matrix $(E|0)$, there is a unique solution associated with every number x_k; while in the equations associated to the matrix $(F|0)$, x_k must be zero to be a solution. This argument contradicts the fact that the $(E|0)$ equations and the $(F|0)$ equations have the same solutions. So the pivots of F must also occur in columns $1, \ldots, \ell$, and the equations associated to F must have the form:

$$\begin{array}{rcl} x_1 & = & -\hat{a}_{1,\ell+1}x_{\ell+1} - \cdots - \hat{a}_{1,n}x_n \\ x_2 & = & -\hat{a}_{2,\ell+1}x_{\ell+1} - \cdots - \hat{a}_{2,n}x_n \\ \vdots & & \vdots \\ x_\ell & = & -\hat{a}_{\ell,\ell+1}x_{\ell+1} - \cdots - \hat{a}_{\ell,n}x_n \end{array} \tag{2.6.2}$$

where $\hat{a}_{i,j}$ are scalars.

To complete this proof, we show that $a_{i,j} = \hat{a}_{i,j}$. These equalities are verified as follows. There is just one solution to each system (2.6.1) and (2.6.2) of the form

$$x_{\ell+1} = 1,\ x_{\ell+2} = \cdots = x_n = 0.$$

These solutions are

$$(-a_{1,\ell+1}, \ldots, -a_{\ell,\ell+1}, 1, 0, \cdots, 0)$$

for (2.6.1) and

$$(-\hat{a}_{1,\ell+1}, \ldots, -\hat{a}_{\ell,\ell+1}, 1, 0 \cdots, 0)$$

for (2.6.2). It follows that $a_{j,\ell+1} = \hat{a}_{j,\ell+1}$ for $j = 1, \ldots, \ell$. Complete this proof by repeating this argument. Just inspect solutions of the form

$$x_{\ell+1} = 0,\ x_{\ell+2} = 1,\ x_{\ell+3} = \cdots = x_n = 0$$

through

$$x_{\ell+1} = \cdots = x_{n-1} = 0,\ x_n = 1.$$

3 Matrices and Linearity

In this chapter we take the first step in abstracting vectors and matrices to mathematical objects that are more than just arrays of numbers. We begin the discussion in Section 3.1 by introducing the multiplication of a matrix times a vector. Matrix multiplication simplifies the way in which we write systems of linear equations and is the way by which we view matrices as mappings. This latter point is discussed in Section 3.2.

The mappings that are produced by matrix multiplication are special and are called *linear mappings*. Some properties of linear maps are discussed in Section 3.3. One consequence of linearity is the *principle of superposition* that enables solutions to systems of linear equations to be built out of simpler solutions. This principle is discussed in Section 3.4.

In Section 3.5 we introduce multiplication of two matrices and discuss properties of this multiplication in Section 3.6. Matrix multiplication is defined in terms of composition of linear mappings which leads to an explicit formula for matrix multiplication. This dual role of multiplication of two matrices — first by formula and second as composition — enables us to solve linear equations in a conceptual way as well as in an algorithmic way. The conceptual way of solving linear equations is through the use of matrix inverses (or inverse mappings) which is described in Section 3.7. In this section we also present important properties of matrix inversion and a method of computation of matrix inverses. There is a simple formula for computing inverses of 2×2 matrices based on determinants. The chapter ends with a discussion of determinants of 2×2 matrices in Section 3.8.

3.1 Matrix Multiplication of Vectors

In Chapter 2 we discussed how matrices appear when solving systems of m linear equations in n unknowns. Given the system

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots & & \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m \end{array}, \tag{3.1.1}$$

we saw that all relevant information is contained in the $m \times n$ matrix of coefficients

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

and the m vector

$$b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Matrices Times Vectors We motivate multiplication of a matrix times a vector just as a notational advance that simplifies the presentation of the linear systems. It is, however, much more than that. This concept of multiplication allows us to think of matrices as mappings and these mappings tell us much about the structure of solutions to linear systems. But first we discuss the notational advantage.

Multiplying an $m \times n$ matrix A times an n vector x produces an m vector, as follows:

$$Ax = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + \cdots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \end{pmatrix}. \tag{3.1.2}$$

For example, when $m = 2$ and $n = 3$, then the product is a 2-vector

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \end{pmatrix}. \tag{3.1.3}$$

As a specific example, compute

$$\begin{aligned} \begin{pmatrix} 2 & 3 & -1 \\ 4 & 1 & 5 \end{pmatrix} \begin{pmatrix} 2 \\ -3 \\ 4 \end{pmatrix} &= \begin{pmatrix} 2 \cdot 2 & + & 3 \cdot (-3) & + & (-1) \cdot 4 \\ 4 \cdot 2 & + & 1 \cdot (-3) & + & 5 \cdot 4 \end{pmatrix} \\ &= \begin{pmatrix} -9 \\ 25 \end{pmatrix}. \end{aligned}$$

Using (3.1.2) we have a compact notation for writing systems of linear equations. For example, using a special instance of (3.1.3),

$$\begin{pmatrix} 2 & 3 & -1 \\ 4 & 1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2x_1 + 3x_2 - x_3 \\ 4x_1 + x_2 + 5x_3 \end{pmatrix}.$$

In this notation we can write the system of two linear equations in three unknowns

$$\begin{array}{ccccccc} 2x_1 & + & 3x_2 & - & x_3 & = & 2 \\ 4x_1 & + & x_2 & + & 5x_3 & = & -1 \end{array}$$

as the matrix equation

$$\begin{pmatrix} 2 & 3 & -1 \\ 4 & 1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$$

Indeed, the general system of linear equations (3.1.1) can be written in matrix form using matrix multiplication as

$$Ax = b$$

where A is the $m \times n$ matrix of coefficients, x is the n vector of unknowns, and b is the m vector of constants on the right hand side of (3.1.1).

Matrices Times Vectors in MATLAB We have already seen how to define matrices and vectors in MATLAB. Now we show how to multiply a matrix times a vector using MATLAB.

Load the matrix A

$$A = \begin{pmatrix} 5 & -4 & 3 & -6 & 2 \\ 2 & -4 & -2 & -1 & 1 \\ 1 & 2 & 1 & -5 & 3 \\ -2 & -1 & -2 & 1 & -1 \\ 1 & -6 & 1 & 1 & 4 \end{pmatrix} \qquad (3.1.4^*)$$

and the vector x

$$x = \begin{pmatrix} -1 \\ 2 \\ 1 \\ -1 \\ 3 \end{pmatrix} \qquad (3.1.5^*)$$

into MATLAB by typing

```
e3_1_4
e3_1_5
```

The multiplication Ax can be performed by typing

```
b = A*x
```

and the result should be

```
b =
     2
    -8
    18
    -6
    -1
```

We may verify this result by solving the system of linear equations $Ax = b$. Indeed if we type

```
A\b
```

then we get the vector x back as the answer.

Exercises

3.2 Matrix Mappings

Having illustrated the notational advantage of using matrices and matrix multiplication, we now begin to discuss why there is also a *conceptual advantage* to matrix multiplication, a conceptual advantage that will help us to understand how systems of linear equations and linear differential equations may be solved.

Matrix multiplication allows us to view $m \times n$ matrices as mappings from $\mathbb{R}^n$ to $\mathbb{R}^m$. Let A be an $m \times n$ matrix and let x be an n vector. Then

$$x \mapsto Ax$$

defines a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$.

The simplest example of a matrix mapping is given by 1×1 matrices. Matrix mappings defined from $\mathbb{R} \to \mathbb{R}$ are

$$x \mapsto ax$$

where a is a real number. Note that the graph of this function is just a straight line through the origin (with slope a). From this example we see that matrix mappings are very special mappings indeed. In higher dimensions, matrix mappings provide a richer set of mappings; we explore here *planar* mappings — mappings of the plane into itself — using MATLAB graphics and the program map.

The simplest planar matrix mappings are the *dilatations*. Let $A = cI_2$ where $c > 0$ is a scalar. When $c < 1$ vectors are contracted by a factor of c and and these mappings are examples of *contractions*. When $c > 1$ vectors are stretched or expanded by a factor of c and these dilatations are examples of *expansions*. We now explore some more complicated planar matrix mappings.

The next planar motions that we study are those given by the matrices

$$A = \begin{pmatrix} \lambda & 0 \\ 0 & \mu \end{pmatrix}.$$

Here the matrix mapping is given by $(x, y) \mapsto (\lambda x, \mu y)$; that is, a mapping that independently stretches and/or contracts the x and y coordinates. Even these simple looking mappings can move objects in the plane in a somewhat complicated fashion.

The Program map We use MATLAB to explore planar matrix mappings using the program `map`. In MATLAB type the command

```
map
```

and a window appears labeled Map. The 2×2 matrix

$$\cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{3.2.1}$$

has been pre-entered. Click on the Custom button. In the Icons menu click on an icon — say Dog — and a blue 'Dog' will appear in the graphing window. Next click on the Iterate button and a new version of the Dog will appear in yellow —the yellow Dog is just rotated about the origin counterclockwise by 90° from the blue dog. Indeed, the matrix (3.2.1) rotates the plane counterclockwise by 90°. To verify this statement click on Iterate again and see that the yellow dog rotates 90° counterclockwise into the magenta dog. Of course, the magenta dog is rotated 180° from the original blue dog. Clicking on Iterate once more produces a fourth dog — this one in cyan. Finally, one more click on the Iterate button will rotate the cyan dog into a red dog that exactly covers the original blue dog.

Other matrices will produce different motions of the plane. Click on the Reset button. Then either push the Custom button, type the entries in the matrix, and click on the Iterate button; or choose one of the pre-assigned matrices listed in the Gallery menu and click on the Iterate button. For example, clicking on the Contracting rotation button recalls the matrix

$$\begin{pmatrix} 0.3 & -0.8 \\ 0.8 & 0.3 \end{pmatrix}$$

This matrix rotates the plane through an angle of approximately 69.4° counterclockwise and contracts the plane by a factor of approximately 0.85. Now click on Dog in the Icons menu to bring up the blue dog again. Repeated clicking on Iterate rotates and contracts the dog so that dogs in a cycling set of colors slowly converge towards the origin in a spiral of dogs.[4]

Rotations Rotating the plane counterclockwise through an angle θ is a motion given by a matrix mapping. We show that the matrix that performs this rotation is:

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{3.2.2}$$

To verify that R_θ rotates the plane counterclockwise through angle θ, let v_φ be the unit vector whose angle from the horizontal is φ; that is, $v_\varphi = (\cos\varphi, \sin\varphi)$. We can write every vector in $\mathbb{R}^2$ as rv_φ for some number $r \geq 0$. Using the trigonometric identities for the cosine and sine of the sum of two angles, we have:

$$\begin{aligned} R_\theta(rv_\varphi) &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} r\cos\varphi \\ r\sin\varphi \end{pmatrix} \\ &= \begin{pmatrix} r\cos\theta\cos\varphi - r\sin\theta\sin\varphi \\ r\sin\theta\cos\varphi + r\cos\theta\sin\varphi \end{pmatrix} \\ &= r\begin{pmatrix} \cos(\theta+\varphi) \\ \sin(\theta+\varphi) \end{pmatrix} \\ &= rv_{\varphi+\theta}. \end{aligned}$$

This calculation shows that R_θ rotates every vector in the plane counterclockwise through angle θ.

It follows from (3.2.2) that $R_{180^\circ} = -I_2$. So rotating a vector in the plane by 180° is the same as reflecting the vector through the origin. It also follows that the movement associated with the linear map $x \mapsto -cx$ where $x \in \mathbb{R}^2$ and $c > 0$ may be thought of as a dilatation ($x \mapsto cx$) followed by rotation through 180° ($x \mapsto -x$).

[4] When using the program map first choose an Icon (or Vector), second choose a Matrix from the Gallery (or a Custom matrix), and finally click on Iterate. Then Iterate again or Reset to start over.

We claim that combining dilatations with general rotations produces spirals. Consider the matrix

$$S = \begin{pmatrix} c\cos\theta & -c\sin\theta \\ c\sin\theta & c\cos\theta \end{pmatrix} = cR_\theta$$

where $c < 1$. Then a calculation similar to the previous one shows that

$$S(rv_\varphi) = c(rv_{\varphi+\theta}).$$

So S rotates vectors in the plane while contracting them by the factor c. Thus, multiplying a vector repeatedly by S spirals that vector into the origin. The example that we just considered while using map is

$$\begin{pmatrix} 0.3 & -0.8 \\ 0.8 & 0.3 \end{pmatrix} \cong \begin{pmatrix} 0.85\cos(69.4^\circ) & -0.85\sin(69.4^\circ) \\ 0.85\sin(69.4^\circ) & 0.85\cos(69.4^\circ) \end{pmatrix},$$

which is an example of S with $c = 0.85$ and $\theta = 69.4^\circ$.

A Notation for Matrix Mappings We reinforce the idea that matrices are mappings by introducing a notation for the mapping associated with an $m \times n$ matrix A. Define

$$L_A : \mathbb{R}^n \to \mathbb{R}^m$$

by

$$L_A(x) = Ax,$$

for every $x \in \mathbb{R}^n$.

There are two special matrices: the $m \times n$ zero matrix O all of whose entries are 0 and the $n \times n$ identity matrix I_n whose diagonal entries are 1 and whose off diagonal entries are 0. For instance,

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The mappings associated with these special matrices are also special. Let x be an n vector. Then

$$Ox = 0, \tag{3.2.3}$$

where the 0 on the right hand side of (3.2.3) is the m vector all of whose entries are 0. The mapping L_O is the *zero mapping* — the mapping that maps every vector x to 0.

Similarly,

$$I_n x = x$$

for every vector x. It follows that

$$L_{I_n}(x) = x$$

is the *identity mapping*, since it maps every vector to itself. It is for this reason that the matrix I_n is called the $n \times n$ *identity matrix*.

Exercises

3.3 Linearity

We begin by recalling the vector operations of addition and scalar multiplication. Given two n vectors, vector addition is defined by

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{pmatrix}.$$

Multiplication of a scalar times a vector is defined by

$$c\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} cx_1 \\ \vdots \\ cx_n \end{pmatrix}.$$

Using (3.1.2) we can check that matrix multiplication satisfies

$$\begin{array}{rcll} A(x+y) & = & Ax + Ay & (3.3.1) \\ A(cx) & = & c(Ax). & (3.3.2) \end{array}$$

Using MATLAB we can also verify that the identities (3.3.1) and (3.3.2) are valid for some particular choices of x, y, c and A. For example, let $c = 5$ and

$$A = \begin{pmatrix} 2 & 3 & 4 & 1 \\ 1 & 1 & 2 & 3 \end{pmatrix}, \quad x = \begin{pmatrix} 1 \\ 5 \\ 4 \\ 3 \end{pmatrix}, \quad y = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 4 \end{pmatrix}.$$

(3.3.3*)

Typing `e3_3_3` enters this information into MATLAB. Now type

```
z1 = A*(x+y)
z2 = A*x + A*y
```

and compare `z1` and `z2`. The fact that they are both equal to

$$\begin{pmatrix} 35 \\ 33 \end{pmatrix}$$

verifies (3.3.1) in this case. Similarly, type

```
w1 = A*(c*x)
w2 = c*(A*x)
```

and compare `w1` and `w2` to verify (3.3.2).

The central idea in linear algebra is the notion of *linearity*.

Definition 3.3.1. A mapping $L : \mathbb{R}^n \to \mathbb{R}^m$ is *linear* if

(a) $L(x + y) = L(x) + L(y)$ for all $x, y \in \mathbb{R}^n$.

(b) $L(cx) = cL(x)$ for all $x \in \mathbb{R}^n$ and all scalars $c \in \mathbb{R}$.

To better understand the meaning of Definition 3.3.1(a,b), we verify these conditions for the mapping $L : \mathbb{R}^2 \to \mathbb{R}^2$ defined by

$$L(x) = (x_1 + 3x_2, 2x_1 - x_2), \qquad (3.3.4)$$

where $x = (x_1, x_2) \in \mathbb{R}^2$. To verify Definition 3.3.1(a), let $y = (y_1, y_2) \in \mathbb{R}^2$. Then

$$\begin{array}{rcl} L(x+y) & = & L(x_1 + y_1, x_2 + y_2) \\ & = & ((x_1 + y_1) + 3(x_2 + y_2), 2(x_1 + y_1) - (x_2 + y_2)) \\ & = & (x_1 + y_1 + 3x_2 + 3y_2, 2x_1 + 2y_1 - x_2 - y_2). \end{array}$$

On the other hand,

$$\begin{array}{rcl} L(x) + L(y) & = & (x_1 + 3x_2, 2x_1 - x_2) + (y_1 + 3y_2, 2y_1 - y_2) \\ & = & (x_1 + 3x_2 + y_1 + 3y_2, 2x_1 - x_2 + 2y_1 - y_2). \end{array}$$

Hence

$$L(x + y) = L(x) + L(y)$$

for every pair of vectors x and y in $\mathbb{R}^2$.

Similarly, to verify Definition 3.3.1(b), let $c \in \mathbb{R}$ be a scalar and compute

$$L(cx) = L(cx_1, cx_2) = ((cx_1) + 3(cx_2), 2(cx_1) - (cx_2)).$$

Then compute

$$cL(x) = c(x_1+3x_2, 2x_1-x_2) = (c(x_1+3x_2), c(2x_1-x_2)),$$

from which it follows that

$$L(cx) = cL(x)$$

for every vector $x \in \mathbb{R}^2$ and every scalar $c \in \mathbb{R}$. Thus L is a linear mapping.

In fact, the mapping (3.3.4) is a matrix mapping and could have been written in the form

$$L(x) = \begin{pmatrix} 1 & 3 \\ 2 & -1 \end{pmatrix} x.$$

Hence the linearity of L could have been checked using identities (3.3.1) and (3.3.2). Indeed, matrix mappings are always linear mappings, as we now discuss.

Matrix Mappings are Linear Mappings Let A be an $m \times n$ matrix and recall that the matrix mapping $L_A : \mathbb{R}^n \to \mathbb{R}^m$ is defined by $L_A(x) = Ax$. We may rewrite (3.3.1) and (3.3.2) using this notation as

$$\begin{aligned} L_A(x+y) &= L_A(x) + L_A(y) \\ L_A(cx) &= cL_A(x). \end{aligned}$$

Thus all matrix mappings are linear mappings. We will show that all linear mappings are matrix mappings (see Theorem 3.3.5). But first we discuss linearity in the simplest context of mappings from $\mathbb{R} \to \mathbb{R}$.

Linear and Nonlinear Mappings of $\mathbb{R} \to \mathbb{R}$ Note that 1×1 matrices are just scalars $A = (a)$. It follows from (3.3.1) and (3.3.2) that we have shown that the matrix mappings $L_A(x) = ax$ are all linear, though this point could have been verified directly. Before showing that these are all the linear mappings of $\mathbb{R} \to \mathbb{R}$, we focus on examples of functions of $\mathbb{R} \to \mathbb{R}$ that are *not* linear.

Examples of Mappings that are Not Linear

- $f(x) = x^2$. Calculate
  $$f(x+y) = (x+y)^2 = x^2 + 2xy + y^2$$
  while
  $$f(x) + f(y) = x^2 + y^2.$$
  The two expressions are not equal and $f(x) = x^2$ is not linear.

- $f(x) = e^x$. Calculate
  $$f(x+y) = e^{x+y} = e^x e^y$$
  while
  $$f(x) + f(y) = e^x + e^y.$$
  The two expressions are not equal and $f(x) = e^x$ is not linear.

- $f(x) = \sin x$. Recall that
  $$f(x+y) = \sin(x+y) = \sin x \cos y + \cos x \sin y$$
  while
  $$f(x) + f(y) = \sin x + \sin y.$$
  The two expressions are not equal and $f(x) = \sin x$ is not linear.

Linear Functions of One Variable Suppose we take the opposite approach and ask what functions of $\mathbb{R} \to \mathbb{R}$ are linear. Observe that if $L : \mathbb{R} \to \mathbb{R}$ is linear, then

$$L(x) = L(x \cdot 1).$$

Since we are looking at the special case of linear mappings on $\mathbb{R}$, we note that x is a real number as well as a vector. Thus we can use Definition 3.3.1(b) to observe that

$$L(x \cdot 1) = xL(1).$$

So if we let $a = L(1)$, then we see that

$$L(x) = ax.$$

Thus linear mappings of $\mathbb{R}$ into $\mathbb{R}$ are very special mappings indeed; they are all scalar multiples of the identity mapping.

All Linear Mappings are Matrix Mappings We end this section by proving that every linear mapping is given by matrix multiplication. But first we state and prove two lemmas. There is a standard set of vectors that is used over and over again in linear algebra, which we now define.

Definition 3.3.2. Let j be an integer between 1 and n. The n-vector e_j is the vector that has a 1 in the j^{th} entry and zeros in all other entries.

Lemma 3.3.3. *Let $L_1 : \mathbb{R}^n \to \mathbb{R}^m$ and $L_2 : \mathbb{R}^n \to \mathbb{R}^m$ be linear mappings. Suppose that $L_1(e_j) = L_2(e_j)$ for every $j = 1, \ldots, n$. Then $L_1 = L_2$.*

Proof Let $x = (x_1, \ldots, x_n)$ be a vector in $\mathbb{R}^n$. Then

$$x = x_1e_1 + \cdots + x_ne_n.$$

Linearity of L_1 and L_2 implies that

$$\begin{aligned} L_1(x) &= x_1L_1(e_1) + \cdots + x_nL_1(e_n) \\ &= x_1L_2(e_1) + \cdots + x_nL_2(e_n) \\ &= L_2(x). \end{aligned}$$

Since $L_1(x) = L_2(x)$ for all $x \in \mathbb{R}^n$, it follows that $L_1 = L_2$. ■

Lemma 3.3.4. *Let A be an $m \times n$ matrix. Then Ae_j is the j^{th} column of A.*

Proof Recall the definition of matrix multiplication given in (3.1.2). In that formula, just set x_i equal to zero for all $i \neq j$ and set $x_j = 1$. ■

Theorem 3.3.5. *Let $L : \mathbb{R}^n \to \mathbb{R}^m$ be a linear mapping. Then there exists an $m \times n$ matrix A such that $L = L_A$.*

Proof There are two steps to the proof: determine the matrix A and verify that $L_A = L$.

Let A be the matrix whose j^{th} column is $L(e_j)$. By Lemma 3.3.4 $L(e_j) = Ae_j$; that is, $L(e_j) = L_A(e_j)$. Lemma 3.3.3 implies that $L = L_A$. ■

Theorem 3.3.5 provides a simple way of showing that

$$L(0) = 0$$

for any linear map L. Indeed, $L(0) = L_A(0) = A0 = 0$ for some matrix A. (This fact can also be proved directly from the definition of linear mapping.)

Using Theorem 3.3.5 to Find Matrices Associated to Linear Maps The proof of Theorem 3.3.5 shows that the j^{th} column of the matrix A associated to a linear mapping L is $L(e_j)$ viewed as a column vector. As an example, let $L : \mathbb{R}^2 \to \mathbb{R}^2$ be rotation clockwise through 90°. Geometrically, it is easy to see that

$$L(e_1) = L\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

and

$$L(e_2) = L\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Since we know that rotations are linear maps, it follows that the matrix A associated to the linear map L is:

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

Additional examples of linear mappings whose associated matrices can be found using Theorem 3.3.5 are given in Exercises **??** – **??**.

Exercises

3.4 The Principle of Superposition

The principle of superposition is just a restatement of the fact that matrix mappings are linear. Nevertheless, this restatement is helpful when trying to understand the structure of solutions to systems of linear equations.

Homogeneous Equations A system of linear equations is *homogeneous* if it has the form

$$Ax = 0, \tag{3.4.1}$$

where A is an $m \times n$ matrix and $x \in \mathbb{R}^n$. Note that homogeneous systems are consistent since $0 \in \mathbb{R}^n$ is always a solution, that is, $A(0) = 0$.

The *principle of superposition* makes two assertions:

- Suppose that y and z in $\mathbb{R}^n$ are solutions to (3.4.1) (that is, suppose that $Ay = 0$ and $Az = 0$); then $y + z$ is a solution to (3.4.1).
- Suppose that c is a scalar; then cy is a solution to (3.4.1).

The principle of superposition is proved using the linearity of matrix multiplication. Calculate

$$A(y+z) = Ay + Az = 0 + 0 = 0$$

to verify that $y + z$ is a solution, and calculate

$$A(cy) = c(Ay) = c \cdot 0 = 0$$

to verify that cy is a solution.

We see that solutions to homogeneous systems of linear equations always satisfy the general property of superposition: sums of solutions are solutions and scalar multiples of solutions are solutions.

We illustrate this principle by explicitly solving the system of equations

$$\begin{pmatrix} 1 & 2 & -1 & 1 \\ 2 & 5 & -4 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Use row reduction to show that the matrix

$$\begin{pmatrix} 1 & 2 & -1 & 1 \\ 2 & 5 & -4 & -1 \end{pmatrix}$$

is row equivalent to

$$\begin{pmatrix} 1 & 0 & 3 & 7 \\ 0 & 1 & -2 & -3 \end{pmatrix}$$

which is in reduced echelon form. Recall, using the methods of Section 2.3, that every solution to this linear system has the form

$$\begin{pmatrix} -3x_3 - 7x_4 \\ 2x_3 + 3x_4 \\ x_3 \\ x_4 \end{pmatrix} = x_3 \begin{pmatrix} -3 \\ 2 \\ 1 \\ 0 \end{pmatrix} + x_4 \begin{pmatrix} -7 \\ 3 \\ 0 \\ 1 \end{pmatrix}.$$

Superposition is verified again by observing that the form of the solutions is preserved under vector addition and scalar multiplication. For instance, suppose that

$$\alpha_1 \begin{pmatrix} -3 \\ 2 \\ 1 \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} -7 \\ 3 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \beta_1 \begin{pmatrix} -3 \\ 2 \\ 1 \\ 0 \end{pmatrix} + \beta_2 \begin{pmatrix} -7 \\ 3 \\ 0 \\ 1 \end{pmatrix}$$

are two solutions. Then the sum has the form

$$\gamma_1 \begin{pmatrix} -3 \\ 2 \\ 1 \\ 0 \end{pmatrix} + \gamma_2 \begin{pmatrix} -7 \\ 3 \\ 0 \\ 1 \end{pmatrix}$$

where $\gamma_j = \alpha_j + \beta_j$.

We have actually proved more than superposition. We have shown in this example that every solution is a superposition of just two solutions

$$\begin{pmatrix} -3 \\ 2 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -7 \\ 3 \\ 0 \\ 1 \end{pmatrix}.$$

Inhomogeneous Equations The linear system of m equations in n unknowns is written as

$$Ax = b$$

where A is an $m \times n$ matrix, $x \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$. This system is *inhomogeneous* when the vector b is nonzero. Note that if $y, z \in \mathbb{R}^n$ are solutions to the inhomogeneous equation (that is, $Ay = b$ and $Az = b$), then $y - z$ is a solution to the homogeneous equation. That is,

$$A(y - z) = Ay - Az = b - b = 0.$$

For example, let

$$A = \begin{pmatrix} 1 & 2 & 0 \\ -2 & 0 & 1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 3 \\ -1 \end{pmatrix}.$$

Then

$$y = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{and} \quad z = \begin{pmatrix} 3 \\ 0 \\ 5 \end{pmatrix}$$

are both solutions to the linear system $Ax = b$. It follows that

$$y - z = \begin{pmatrix} -2 \\ 1 \\ -4 \end{pmatrix}$$

is a solution to the homogeneous system $Ax = 0$, which can be checked by direct calculation.

Thus we can completely solve the inhomogeneous equation by finding one solution to the inhomogeneous equation and then adding to that solution every solution of the homogeneous equation. More precisely, suppose that we know all of the solutions w to the homogeneous equation $Ax = 0$ and one solution y to the inhomogeneous equation $Ax = b$. Then $y + w$ is another solution to the inhomogeneous equation and *every* solution to the inhomogeneous equation has this form.

An Example of an Inhomogeneous Equation Suppose that we want to find all solutions of $Ax = b$ where

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 0 & 1 & -2 \\ 3 & 3 & -1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix}.$$

Suppose that you are told that $y = (-5, 6, 1)^t$ is a solution of the inhomogeneous equation. (This fact can be verified by a short calculation — just multiply Ay and see that the result equals b.) Next find all solutions to the homogeneous equation $Ax = 0$ by putting A into reduced echelon form. The resulting row echelon form matrix is

$$\begin{pmatrix} 1 & 0 & \frac{5}{3} \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{pmatrix}.$$

Hence we see that the solutions of the homogeneous equation $Ax = 0$ are

$$\begin{pmatrix} -\frac{5}{3}s \\ 2s \\ s \end{pmatrix} = s \begin{pmatrix} -\frac{5}{3} \\ 2 \\ 1 \end{pmatrix}.$$

Combining these results, we conclude that all the solutions of $Ax = b$ are given by

$$\begin{pmatrix} -5 \\ 6 \\ 1 \end{pmatrix} + s \begin{pmatrix} -\frac{5}{3} \\ 2 \\ 1 \end{pmatrix}.$$

Exercises

3.5 Composition and Multiplication of Matrices

The *composition* of two matrix mappings leads to another matrix mapping from which the concept of multiplication of two matrices follows. Matrix multiplication can be introduced by formula, but then the idea is unmotivated and one is left to wonder why matrix multiplication is defined in such a seemingly awkward way.

We begin with the example of 2×2 matrices. Suppose that

$$A = \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & 3 \\ -1 & 4 \end{pmatrix}.$$

We have seen that the mappings

$$x \mapsto Ax \quad \text{and} \quad x \mapsto Bx$$

map 2-vectors to 2-vectors. So we can ask what happens when we compose these mappings. In symbols, we compute

$$L_A \circ L_B(x) = L_A(L_B(x)) = A(Bx).$$

In coordinates, let $x = (x_1, x_2)$ and compute

$$\begin{aligned} A(Bx) &= A\begin{pmatrix} 3x_2 \\ -x_1 + 4x_2 \end{pmatrix} \\ &= \begin{pmatrix} -x_1 + 10x_2 \\ x_1 - x_2 \end{pmatrix}. \end{aligned}$$

It follows that we can rewrite $A(Bx)$ using multiplication of a matrix times a vector as

$$A(Bx) = \begin{pmatrix} -1 & 10 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

In particular, $L_A \circ L_B$ is again a linear mapping, namely L_C, where

$$C = \begin{pmatrix} -1 & 10 \\ 1 & -1 \end{pmatrix}.$$

With this computation in mind, we define the product

$$AB = \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 3 \\ -1 & 4 \end{pmatrix} = \begin{pmatrix} -1 & 10 \\ 1 & -1 \end{pmatrix}.$$

Using the same approach we can derive a formula for matrix multiplication of 2×2 matrices. Suppose

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}.$$

Then

$$\begin{aligned} A(Bx) &= A\begin{pmatrix} b_{11}x_1 + b_{12}x_2 \\ b_{21}x_1 + b_{22}x_2 \end{pmatrix} \\ &= \begin{pmatrix} a_{11}(b_{11}x_1 + b_{12}x_2) + a_{12}(b_{21}x_1 + b_{22}x_2) \\ a_{21}(b_{11}x_1 + b_{12}x_2) + a_{22}(b_{21}x_1 + b_{22}x_2) \end{pmatrix} \\ &= \begin{pmatrix} (a_{11}b_{11} + a_{12}b_{21})x_1 + (a_{11}b_{12} + a_{12}b_{22})x_2 \\ (a_{21}b_{11} + a_{22}b_{21})x_1 + (a_{21}b_{12} + a_{22}b_{22})x_2 \end{pmatrix} \\ &= \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \end{aligned}$$

Hence, for 2×2 matrices, we see that composition of matrix mappings defines the matrix multiplication

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

to be

$$\begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}. \tag{3.5.1}$$

Formula (3.5.1) may seem a bit formidable, but it does have structure. Suppose A and B are 2×2 matrices, then the entry of

$$C = AB$$

in the i^{th} row, j^{th} column may be written as

$$a_{i1}b_{1j} + a_{i2}b_{2j} = \sum_{k=1}^{2} a_{ik}b_{kj}.$$

We shall see that an analog of this formula is available for matrix multiplications of all sizes. But to derive this formula, it is easier to develop matrix multiplication abstractly.

Lemma 3.5.1. *Let $L_1 : \mathbb{R}^n \to \mathbb{R}^m$ and $L_2 : \mathbb{R}^p \to \mathbb{R}^n$ be linear mappings. Then $L = L_1 \circ L_2 : \mathbb{R}^p \to \mathbb{R}^m$ is a linear mapping.*

Proof Compute

$$\begin{aligned} L(x+y) &= L_1 \circ L_2(x+y) \\ &= L_1(L_2(x) + L_2(y)) \\ &= L_1(L_2(x)) + L_1(L_2(y)) \\ &= L_1 \circ L_2(x) + L_1 \circ L_2(y) \\ &= L(x) + L(y). \end{aligned}$$

Similarly, compute $L_1 \circ L_2(cx) = cL_1 \circ L_2(x)$. ■

We apply Lemma 3.5.1 in the following way. Let A be an $m \times n$ matrix and let B be an $n \times p$ matrix. Then $L_A : \mathbb{R}^n \to \mathbb{R}^m$ and $L_B : \mathbb{R}^p \to \mathbb{R}^n$ are linear mappings, and the mapping $L = L_A \circ L_B : \mathbb{R}^p \to \mathbb{R}^m$ is defined and linear. Theorem 3.3.5 implies that there is an $m \times p$ matrix C such that $L = L_C$. Abstractly, we define the *matrix product* AB to be C.

> *Note that the matrix product AB is defined only when the number of columns of A is equal to the number of rows of B.*

Calculating the Product of Two Matrices Next we discuss how to calculate the product of matrices; this discussion generalizes our discussion of the product of 2×2 matrices. Lemma 3.3.4 tells how to compute $C = AB$. The j^{th} column of the matrix product is just

$$Ce_j = A(Be_j),$$

where $B_j \equiv Be_j$ is the j^{th} column of the matrix B. Therefore,

$$C = (AB_1 | \cdots | AB_p). \tag{3.5.2}$$

Indeed, the $(i,j)^{th}$ entry of C is the i^{th} entry of AB_j, that is, the i^{th} entry of

$$A \begin{pmatrix} b_{1j} \\ \vdots \\ b_{nj} \end{pmatrix} = \begin{pmatrix} a_{11}b_{1j} + \cdots + a_{1n}b_{nj} \\ \vdots \\ a_{m1}b_{1j} + \cdots + a_{mn}b_{nj} \end{pmatrix}.$$

It follows that the entry c_{ij} of C in the i^{th} row and j^{th} column is

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}. \tag{3.5.3}$$

We can interpret (3.5.3) in the following way. To calculate c_{ij}: multiply the entries of the i^{th} row of A with the corresponding entries in the j^{th} column of B and add the results. This interpretation reinforces the idea that for the matrix product AB to be defined, the number of columns in A must equal the number of rows in B.

For example, we now perform the following multiplication:

$$\begin{aligned} & \begin{pmatrix} 2 & 3 & 1 \\ 3 & -1 & 2 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ 3 & 1 \\ -1 & 4 \end{pmatrix} \\ = & \begin{pmatrix} 2 \cdot 1 + 3 \cdot 3 + 1 \cdot (-1) & 2 \cdot (-2) + 3 \cdot 1 + 1 \cdot 4 \\ 3 \cdot 1 + (-1) \cdot 3 + 2 \cdot (-1) & 3 \cdot (-2) + (-1) \cdot 1 + 2 \cdot 4 \end{pmatrix} \\ = & \begin{pmatrix} 10 & 3 \\ -2 & 1 \end{pmatrix}. \end{aligned}$$

Some Special Matrix Products Let A be an $m\times n$ matrix. Then

$$\begin{aligned} OA &= O \\ AO &= O \\ AI_n &= A \\ I_mA &= A \end{aligned}$$

The first two equalities are easily checked using (3.5.3). It is not significantly more difficult to verify the last two equalities using (3.5.3), but we shall verify these equalities using the language of linear mappings, as follows:

$$L_{AI_n}(x) = L_A \circ L_{I_n}(x) = L_A(x),$$

since $L_{I_n}(x) = x$ is the identity map. Therefore $AI_n = A$. A similar proof verifies that $I_mA = A$. Although the verification of these equalities using the notions of linear mappings may appear to be a case of overkill, the next section contains results where these notions truly simplify the discussion.

Exercises

3.6 Properties of Matrix Multiplication

In this section we discuss the facts that matrix multiplication is associative (but not commutative) and that certain distributive properties hold. We also discuss how matrix multiplication is performed in MATLAB .

Matrix Multiplication is Associative

Theorem 3.6.1. *Matrix multiplication is associative. That is, let A be an $m \times n$ matrix, let B be a $n \times p$ matrix, and let C be a $p \times q$ matrix. Then*

$$(AB)C = A(BC).$$

Proof Begin by observing that composition of mappings is always associative. In symbols, let $f : \mathbb{R}^n \to \mathbb{R}^m$, $g : \mathbb{R}^p \to \mathbb{R}^n$, and $h : \mathbb{R}^q \to \mathbb{R}^p$. Then

$$\begin{aligned} f\circ(g\circ h)(x) &= f[(g\circ h)(x)] \\ &= f[g(h(x))] \\ &= (f\circ g)(h(x)) \\ &= [(f\circ g)\circ h](x). \end{aligned}$$

It follows that

$$f\circ(g\circ h) = (f\circ g)\circ h.$$

We can apply this result to linear mappings. Thus

$$L_A\circ(L_B\circ L_C) = (L_A\circ L_B)\circ L_C.$$

Since

$$L_{A(BC)} = L_A\circ L_{BC} = L_A\circ(L_B\circ L_C)$$

and

$$L_{(AB)C} = L_{AB}\circ L_C = (L_A\circ L_B)\circ L_C,$$

it follows that

$$L_{A(BC)} = L_{(AB)C},$$

and

$$A(BC) = (AB)C.$$

■

It is worth convincing yourself that Theorem 3.6.1 has content by verifying by hand that matrix multiplication of 2×2 matrices is associative.

Matrix Multiplication is Not Commutative Although matrix multiplication is associative, it is *not* commutative. This statement is trivially true when the matrix AB is defined while that matrix BA is not. Suppose, for example, that A is a 2×3 matrix and that B is a 3×4 matrix. Then AB is a 2×4 matrix, while the multiplication BA makes no sense whatsoever.

More importantly, suppose that A and B are both $n \times n$ square matrices. Then $AB = BA$ is generally not valid. For example, let

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

Then

$$AB = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad BA = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

So $AB \neq BA$. In certain cases it does happen that $AB = BA$. For example, when $B = I_n$,

$$AI_n = A = I_nA.$$

But these cases are rare.

Additional Properties of Matrix Multiplication Recall that if $A = (a_{ij})$ and $B = (b_{ij})$ are both $m \times n$ matrices, then $A+B$ is the $m\times n$ matrix $(a_{ij} + b_{ij})$. We now enumerate several properties of matrix multiplication.

- Let A and B be $m\times n$ matrices and let C be an $n\times p$ matrix. Then
$$(A+B)C = AC+BC.$$
Similarly, if D is a $q\times m$ matrix, then
$$D(A+B) = DA+DB.$$
So matrix multiplication distributes across matrix addition.

- If α and β are scalars, then
$$(\alpha+\beta)A = \alpha A+\beta A.$$
So addition distributes with scalar multiplication.

- Scalar multiplication and matrix multiplication satisfy:
$$(\alpha A)C = \alpha(AC).$$

Matrix Multiplication and Transposes Let A be an $m\times n$ matrix and let B be an $n\times p$ matrix, so that the matrix product AB is defined and AB is an $m\times p$ matrix. Note that A^t is an $n\times m$ matrix and that B^t is a $p\times n$ matrix, so that in general the product A^tB^t is *not* defined. However, the product B^tA^t is defined and is an $p\times m$ matrix, as is the matrix $(AB)^t$. We claim that
$$(AB)^t = B^tA^t. \tag{3.6.1}$$
We verify this claim by direct computation. The $(i,k)^{th}$ entry in $(AB)^t$ is the $(k,i)^{th}$ entry in AB. That entry is:
$$\sum_{j=1}^n a_{kj}b_{ji}.$$
The $(i,k)^{th}$ entry in B^tA^t is:
$$\sum_{j=1}^n b^t_{ij}a^t_{jk},$$
where a^t_{jk} is the $(j,k)^{th}$ entry in A^t and b^t_{ij} is the $(i,j)^{th}$ entry in B^t. It follows from the definition of transpose that the $(i,k)^{th}$ entry in B^tA^t is:
$$\sum_{j=1}^n b_{ji}a_{kj} = \sum_{j=1}^n a_{kj}b_{ji},$$
which verifies the claim.

Matrix Multiplication in MATLAB Let us now explain how matrix multiplication works in MATLAB. We load the matrices
$$A = \begin{pmatrix} -5 & 2 & 0 \\ -1 & 1 & -4 \\ -4 & 4 & 2 \\ -1 & 3 & -1 \end{pmatrix} \quad\text{and}\quad B = \begin{pmatrix} 2 & -2 & -2 & 5 & 5 \\ 4 & -5 & 1 & -1 & 2 \\ 3 & 2 & 3 & -3 & 3 \end{pmatrix}$$
(3.6.2*)
by typing

```
e3_6_2
```

Now the command `C = A*B` asks MATLAB to compute the matrix C as the product of A and B. We obtain

```
C =
    -2     0    12   -27   -21
   -10   -11    -9     6   -15
    14    -8    18   -30    -6
     7   -15     2    -5    -2
```

Let us confirm this result by another computation. As we have seen above the 4^{th} column of C should be given by the product of A with the 4^{th} column of B. Indeed, if we perform this computation and type

```
A*B(:,4)
```

the result is

```
ans =
   -27
     6
   -30
    -5
```

which is precisely the 4^{th} column of C.

MATLAB also recognizes when a matrix multiplication of two matrices is not defined. For example, the product of the 3×5 matrix B with the 4×3 matrix A is not defined, and if we type `B*A` then we obtain the error message

```
??? Error using ==> *
Inner matrix dimensions must agree.
```

We remark that the size of a matrix A can be seen using the MATLAB command `size`. For example, the command `size(A)` leads to

```
ans =
     4     3
```

reflecting the fact that A is a matrix with four rows and three columns.

Exercises

3.7 Solving Linear Systems and Inverses

When we solve the simple equation

$$ax = b,$$

we do so by dividing by a to obtain

$$x = \frac{1}{a}b.$$

This division works as long as $a \neq 0$.

Writing systems of linear equations as

$$Ax = b$$

suggests that solutions should have the form

$$x = \frac{1}{A}b$$

and the MATLAB command for solving linear systems

```
x=A\b
```

suggests that there is some merit to this analogy.

The following is a better analogy. Multiplication by a has the inverse operation: division by a; multiplying a number x by a and then multiplying the result by $a^{-1} = 1/a$ leaves the number x unchanged (as long as $a \neq 0$). In this sense we should write the solution to $ax = b$ as

$$x = a^{-1}b.$$

For systems of equations $Ax = b$ we wish to write solutions as

$$x = A^{-1}b.$$

In this section we consider the questions: What does A^{-1} mean and when does A^{-1} exist? (Even in one dimension, we have seen that the inverse does not always exist, since $0^{-1} = \frac{1}{0}$ is undefined.)

Invertibility We begin by giving a precise definition of invertibility for square matrices.

Definition 3.7.1. The $n \times n$ matrix A is *invertible* if there is an $n \times n$ matrix B such that

$$AB = I_n \quad \text{and} \quad BA = I_n.$$

The matrix B is called an *inverse* of A. If A is not invertible, then A is *noninvertible* or *singular*.

Geometrically, we can see that some matrices are invertible. For example, the matrix

$$R_{90} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

rotates the plane counterclockwise through 90° and is invertible. The inverse matrix of R_{90} is the matrix that rotates the plane clockwise through 90°. That matrix is:

$$R_{-90} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

This statement can be checked algebraically by verifying that $R_{90}R_{-90} = I_2$ and that $R_{-90}R_{90} = I_2$.

Similarly,

$$B = \begin{pmatrix} 5 & 3 \\ 2 & 1 \end{pmatrix}$$

is an inverse of

$$A = \begin{pmatrix} -1 & 3 \\ 2 & -5 \end{pmatrix},$$

as matrix multiplication shows that $AB = I_2$ and $BA = I_2$. In fact, there is an elementary formula for finding inverses of 2×2 matrices (when they exist); see (3.8.1) in Section 3.8.

On the other hand, not all matrices are invertible. For example, the zero matrix is noninvertible, since $0B = 0$ for any matrix B.

Lemma 3.7.2. *If an $n \times n$ matrix A is invertible, then its inverse is unique and is denoted by A^{-1}.*

Proof Let B and C be $n \times n$ matrices that are inverses of A. Then

$$BA = I_n \quad \text{and} \quad AC = I_n.$$

We use the associativity of matrix multiplication to prove that $B = C$. Compute

$$B = BI_n = B(AC) = (BA)C = I_nC = C.$$

■

We now show how to compute inverses for products of invertible matrices.

Proposition 3.7.3. *Let A and B be two invertible $n \times n$ matrices. Then AB is also invertible and*

$$(AB)^{-1} = B^{-1}A^{-1}.$$

Proof Use associativity of matrix multiplication to compute

$$(AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = AI_nA^{-1} = AA^{-1} = I_n.$$

Similarly,

$$(B^{-1}A^{-1})(AB) = B^{-1}(A^{-1}A)B = B^{-1}B = I_n.$$

Therefore AB is invertible with the desired inverse. ■

Proposition 3.7.4. *Suppose that A is an invertible $n \times n$ matrix. Then A^t is invertible and*

$$(A^t)^{-1} = (A^{-1})^t.$$

Proof We must show that $(A^{-1})^t$ is the inverse of A^t. Identity (3.6.1) implies that

$$(A^{-1})^tA^t = (AA^{-1})^t = (I_n)^t = I_n,$$

and

$$A^t(A^{-1})^t = (A^{-1}A)^t = (I_n)^t = I_n.$$

Therefore, $(A^{-1})^t$ is the inverse of A^t, as claimed. ■

Invertibility and Unique Solutions Next we discuss the implications of invertibility for the solution of the inhomogeneous linear system:

$$Ax = b, \tag{3.7.1}$$

where A is an $n \times n$ matrix and $b \in \mathbb{R}^n$.

Proposition 3.7.5. *Let A be an invertible $n \times n$ matrix and let b be in $\mathbb{R}^n$. Then the system of linear equations (3.7.1) has a unique solution.*

Proof We can solve the linear system (3.7.1) by setting

$$x = A^{-1}b. \tag{3.7.2}$$

This solution is easily verified by calculating

$$Ax = A(A^{-1}b) = (AA^{-1})b = I_nb = b.$$

Next, suppose that x is a solution to (3.7.1). Then

$$x = I_nx = (A^{-1}A)x = A^{-1}(Ax) = A^{-1}b.$$

So $A^{-1}b$ is the only possible solution. ■

Corollary 3.7.6. *An invertible matrix is row equivalent to I_n.*

Proof Let A be an invertible $n \times n$ matrix. Proposition 3.7.5 states that the system of linear equations $Ax = b$ has a unique solution. Chapter 2, Corollary 2.4.8 states that A is row equivalent to I_n. ■

The converse of Corollary 3.7.6 is also valid.

Proposition 3.7.7. *An $n \times n$ matrix A that is row equivalent to I_n is invertible.*

Proof Form the $n \times 2n$ matrix $M = (A|I_n)$. Since A is row equivalent to I_n, there is a sequence of elementary row operations so that M is row equivalent to

$(I_n|B)$. Eliminating all columns from the right half of M except the j^{th} column yields the matrix $(A|e_j)$. The same sequence of elementary row operations states that the matrix $(A|e_j)$ is row equivalent to $(I_n|B_j)$ where B_j is the j^{th} column of B. It follows that B_j is the solution to the system of linear equations $Ax = e_j$ and that the matrix product

$$AB = (AB_1|\cdots|AB_n) = (e_1|\cdots|e_n) = I_n.$$

So $AB = I_n$.

We claim that $BA = I_n$ and hence that A is invertible. To verify this claim form the $n \times 2n$ matrix $N = (I_n|A)$. Using the same sequence of elementary row operations again shows that N is row equivalent to $(B|I_n)$. By construction the matrix B is row equivalent to I_n. Therefore, there is a unique solution to the system of linear equations $Bx = e_j$. Now eliminating all columns except the j^{th} from the right hand side of the matrix $(B|I_n)$ shows that the solution to the system of linear equations $Bx = e_j$ is just A_j, where A_j is the j^{th} column of A. It follows that

$$BA = (BA_1|\cdots|BA_n) = (e_1|\cdots|e_n) = I_n.$$

Hence $BA = I_n$. ■

Theorem 3.7.8. *Let A be an $n \times n$ matrix. Then the following are equivalent:*

(a) A is invertible.

(b) The equation $Ax = b$ has a unique solution for each $b \in \mathbb{R}^n$.

(c) The only solution to $Ax = 0$ is $x = 0$.

(d) A is row equivalent to I_n.

Proof $(a) \Rightarrow (b)$ This implication is just Proposition 3.7.5.

$(b) \Rightarrow (c)$ This implication is straightforward — just take $b = 0$ in (3.7.1).

$(c) \Rightarrow (d)$ This implication is just a restatement of Chapter 2, Corollary 2.4.8.

$(d) \Rightarrow (a)$. This implication is just Proposition 3.7.7. ■

A Method for Computing Inverse Matrices The proof of Proposition 3.7.7 gives a constructive method for finding the inverse of any invertible square matrix.

Theorem 3.7.9. *Let A be an $n \times n$ matrix that is row equivalent to I_n and let M be the $n \times 2n$ augmented matrix*

$$M = (A|I_n). \tag{3.7.3}$$

Then the matrix M is row equivalent to $(I_n|A^{-1})$.

An Example Compute the inverse of the matrix

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}.$$

Begin by forming the 3×6 matrix

$$M = \left(\begin{array}{ccc|ccc} 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 1 & 3 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right).$$

To put M in row echelon form by row reduction, first subtract 3 times the 3^{rd} row from the 2^{nd} row, obtaining

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -3 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right).$$

Second, subtract 2 times the 2^{nd} row from the 1^{st} row, obtaining

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & -2 & 6 \\ 0 & 1 & 0 & 0 & 1 & -3 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right).$$

Theorem 3.7.9 implies that

$$A^{-1} = \begin{pmatrix} 1 & -2 & 6 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{pmatrix},$$

which can be verified by matrix multiplication.

Computing the Inverse Using MATLAB There are two ways that we can compute inverses using MATLAB . Either we can perform the row reduction of (3.7.3) directly or we can use the MATLAB the command `inv`. We illustrate both of these methods. First type `e3_7_4` to recall the matrix

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 1 & 1 \\ 2 & 0 & -1 \end{pmatrix}. \tag{3.7.4*}$$

To perform the row reduction of (3.7.3) we need to form the matrix M. The MATLAB command for generating an $n \times n$ identity matrix is `eye(n)`. Therefore, typing

```
M = [A eye(3)]
```

in MATLAB yields the result

```
M =
     1     2     4     1     0     0
     3     1     1     0     1     0
     2     0    -1     0     0     1
```

Now row reduce M to reduced echelon form as follows. Type

```
M(3,:) = M(3,:) - 2*M(1,:)
M(2,:) = M(2,:) - 3*M(1,:)
```

obtaining

```
M =
     1     2     4     1     0     0
     0    -5   -11    -3     1     0
     0    -4    -9    -2     0     1
```

Next type

```
M(2,:) = M(2,:)/M(2,2)
M(3,:) = M(3,:) + 4*M(2,:)
M(1,:) = M(1,:) - 2*M(2,:)
```

to obtain

```
M =
    1.0000         0   -0.4000   -0.2000    0.4000         0
         0    1.0000    2.2000    0.6000   -0.2000         0
         0         0   -0.2000    0.4000   -0.8000    1.0000
```

Finally, type

```
M(3,:) = M(3,:)/M(3,3)
M(2,:) = M(2,:) - M(2,3)*M(3,:)
M(1,:) = M(1,:) - M(1,3)*M(3,:)
```

to obtain

```
M =
    1.0000         0         0   -1.0000    2.0000   -2.0000
         0    1.0000         0    5.0000   -9.0000   11.0000
         0         0    1.0000   -2.0000    4.0000   -5.0000
```

Thus $C = A^{-1}$ is obtained by extracting the last three columns of M by typing

```
C = M(:,[4 5 6])
```

which yields

```
C =
   -1.0000    2.0000   -2.0000
    5.0000   -9.0000   11.0000
   -2.0000    4.0000   -5.0000
```

You may check that C is the inverse of A by typing `A*C` and `C*A`.

In fact, this entire scheme for computing the inverse of a matrix has been preprogrammed into MATLAB . Just type

```
inv(A)
```

to obtain

```
ans =
   -1.0000    2.0000   -2.0000
    5.0000   -9.0000   11.0000
   -2.0000    4.0000   -5.0000
```

We illustrate again this simple method for computing the inverse of a matrix A. For example, reload the matrix in (3.1.4*) by typing `e3_1_4` and obtaining:

```
A =
     5    -4     3    -6     2
     2    -4    -2    -1     1
     1     2     1    -5     3
    -2    -1    -2     1    -1
     1    -6     1     1     4
```

The command `B = inv(A)` stores the inverse of the matrix A in the matrix B, and we obtain the result

```
B =
   -0.0712    0.2856   -0.0862   -0.4813   -0.0915
   -0.1169    0.0585    0.0690   -0.2324   -0.0660
    0.1462   -0.3231   -0.0862    0.0405    0.0825
   -0.1289    0.0645   -0.1034   -0.2819    0.0555
   -0.1619    0.0810    0.1724   -0.1679    0.1394
```

This computation also illustrates the fact that even when the matrix A has integer entries, the inverse of A usually has noninteger entries.

Let $b = (2, -8, 18, -6, -1)$. Then we may use the inverse $B = A^{-1}$ to compute the solution of $Ax = b$. Indeed if we type

```
b = [2;-8;18;-6;-1];
x = B*b
```

then we obtain

```
x =
   -1.0000
    2.0000
    1.0000
   -1.0000
    3.0000
```

as desired (see (3.1.5*)). With this computation we have confirmed the analytical results of the previous subsections.

Exercises

3.8 Determinants of 2×2 Matrices

There is a simple way for determining whether a 2×2 matrix A is invertible and there is a simple formula for finding A^{-1}. First, we present the formula. Let

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

and suppose that $ad - bc \neq 0$. Then

$$A^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}. \qquad (3.8.1)$$

This is most easily verified by directly applying the formula for matrix multiplication. So A is invertible when $ad - bc \neq 0$. We shall prove below that $ad - bc$ must be nonzero when A is invertible.

From this discussion it is clear that the number $ad - bc$ must be an important quantity for 2×2 matrices. So we define:

Definition 3.8.1. The *determinant* of the 2×2 matrix A is

$$\det(A) = ad - bc. \qquad (3.8.2)$$

Proposition 3.8.2. *As a function on 2×2 matrices, the determinant satisfies the following properties.*

(a) The determinant of an upper triangular matrix is the product of the diagonal elements.

(b) The determinants of a matrix and its transpose are equal.

(c) $\det(AB) = \det(A)\det(B)$.

Proof Both (a) and (b) are easily verified by direct calculation. Property (c) is also verified by direct calculation — but of a more extensive sort. Note that

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} a\alpha + b\gamma & a\beta + b\delta \\ c\alpha + d\gamma & c\beta + d\delta \end{pmatrix}.$$

Therefore,

$$\begin{aligned} \det(AB) &= (a\alpha + b\gamma)(c\beta + d\delta) - (a\beta + b\delta)(c\alpha + d\gamma) \\ &= (ac\alpha\beta + bc\beta\gamma + ad\alpha\delta + bd\gamma\delta) \\ &\quad -(ac\alpha\beta + bc\alpha\delta + ad\beta\gamma + bd\gamma\delta) \\ &= bc(\beta\gamma - \alpha\delta) + ad(\alpha\delta - \beta\gamma) \\ &= (ad - bc)(\alpha\delta - \beta\gamma) \\ &= \det(A)\det(B), \end{aligned}$$

as asserted. ■

Corollary 3.8.3. *A 2×2 matrix A is invertible if and only if* $\det(A) \neq 0$.

Proof If A is invertible, then $AA^{-1} = I_2$. Proposition 3.8.2 implies that

$$\det(A)\det(A^{-1}) = \det(I_2) = 1.$$

Therefore, $\det(A) \neq 0$. Conversely, if $\det(A) \neq 0$, then (3.8.1) implies that A is invertible. ■

Determinants and Area Suppose that v and w are two vectors in $\mathbb{R}^2$ that point in different directions. Then, the set of points

$$z = \alpha v + \beta w \quad \text{where } 0 \leq \alpha, \beta \leq 1$$

is a parallelogram, that we denote by P. We denote the area of P by $|P|$. For example, the unit square S, whose corners are $(0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$, is the parallelogram generated by the unit vectors e_1 and e_2.

Next let A be a 2×2 matrix and let

$$A(P) = \{Az : z \in P\}.$$

It follows from linearity (since $Az = \alpha Av + \beta Aw$) that $A(P)$ is the parallelogram generated by Av and Aw.

Proposition 3.8.4. *Let A be a 2×2 matrix and let S be the unit square. Then*

$$|A(S)| = |\det A|. \tag{3.8.3}$$

Proof Note that $A(S)$ is the parallelogram generated by $u_1 = Ae_1$ and $u_2 = Ae_2$, and u_1 and u_2 are the columns of A. It follows that

$$(\det A)^2 = \det(A^t)\det(A) = \det(A^tA) = \det\begin{pmatrix} u_1^tu_1 & u_1^tu_2 \\ u_2^tu_1 & u_2^tu_2 \end{pmatrix}.$$

Hence

$$(\det A)^2 = \det\begin{pmatrix} ||u_1||^2 & u_1\cdot u_2 \\ u_1\cdot u_2 & ||u_2||^2 \end{pmatrix} = ||u_1||^2||u_2||^2-(u_1\cdot u_2)^2.$$

Recall that (1.4.5) of Chapter 1 states that

$$|P|^2 = ||v||^2||w||^2 - (v\cdot w)^2.$$

where P is the parallelogram generated by v and w. Therefore, $(\det A)^2 = |A(S)|^2$ and (3.8.3) is verified. ■

Theorem 3.8.5. *Let P be a parallelogram in $\mathbb{R}^2$ and let A be a 2×2 matrix. Then*

$$|A(P)| = |\det A||P|. \tag{3.8.4}$$

Proof First note that (3.8.3) a special case of (3.8.4), since $|S| = 1$. Next, let P be the parallelogram generated by the (column) vectors v and w, and let $B = (v|w)$. Then $P = B(S)$. It follows from (3.8.3) that $|P| = |\det B|$. Moreover,

$$\begin{aligned} |A(P)| &= |(AB)(S)| \\ &= |\det(AB)| \\ &= |\det A||\det B| \\ &= |\det A||P|, \end{aligned}$$

as desired. ■

Exercises

4 Solving Linear Differential Equations

The study of linear systems of equations given in Chapter 2 provides one motivation for the study of matrices and linear algebra. Linear constant coefficient systems of ordinary differential equations provide a second motivation for this study. In this chapter we show how the phase space geometry of systems of differential equations motivates the idea of *eigendirections* (or invariant directions) and *eigenvalues* (or growth rates).

We begin this chapter with a discussion of the theory and application of the simplest of linear differential equations, the linear growth equation, $\dot{x} = \lambda x$. In Section 4.1, we solve the linear growth equation and discuss the fact that solutions to differential equations are functions; and we emphasize this point by using MATLAB to graph solutions of x as a function of t. In the optional Section 4.2 we illustrate the applicability of this very simple equation with a discussion of compound interest and a simple population model.

The next two sections introduce planar constant coefficient linear differential equations. In these sections we use the program pplane10 (written by John Polking) that solves numerically planar systems of differential equations. In Section 4.3 we discuss uncoupled systems — two independent one dimensional systems like those presented in Section 4.1 — whose solution geometry in the plane is somewhat more complicated than might be expected. In Section 4.4 we discuss coupled linear systems. Here we illustrate the existence and nonexistence of eigendirections.

In Section 4.5 we show how initial value problems can be solved by building the solution — through the use of superposition as discussed in Section 3.4 — from simpler solutions. These simpler solutions are ones generated from real eigenvalues and eigenvectors — when they exist. In Section 4.6 we develop the theory of *eigenvalues* and *characteristic polynomials* of 2×2 matrices. (The corresponding theory for $n \times n$ matrices is developed in Chapter 7.)

The method for solving planar constant coefficient linear differential equations with real eigenvalues is summarized in Section 4.7. This method is based on the material of Sections 4.5 and 4.6. The complete discussion of the solutions of linear planar systems of differential equations is given in Chapter 6. This discussion is best done after we have introduced the linear algebra concepts of vector subspaces and bases in Chapter 5.

The chapter ends with an optional discussion of *Markov chains* in Section 4.8. Markov chains give a method for analyzing branch processes where at each time unit several outcomes are possible, each with a given probability.

4.1 A Single Differential Equation

Algebraic operations such as addition and multiplication are performed on numbers while the calculus operations of differentiation and integration are performed on functions. Thus algebraic equations (such as $x^2 = 9$) are solved for numbers ($x = \pm 3$) while differential (and integral) equations are solved for functions.

In Chapter 2 we discussed how to solve systems of linear equations, such as

$$\begin{array}{rcl} x_1 + x_2 & = & 2 \\ x_1 - x_2 & = & 4 \end{array}$$

for numbers

$$x_1 = 3 \quad \text{and} \quad x_2 = -1,$$

while in this chapter we discuss how to solve some linear systems of differential equations for functions.

Solving a single linear equation in one unknown x is a simple task. For example, solve

$$2x = 4$$

for $x = 2$. Solving a single differential equation in one unknown function $x(t)$ is far from trivial.

Integral Calculus as a Differential Equation Mathematically, the simplest type of differential equation is:

$$\frac{dx}{dt}(t) = f(t) \tag{4.1.1}$$

where f is some continuous function. In words, this equation asks us to find all functions $x(t)$ whose derivative is $f(t)$. The fundamental theorem of calculus tells us the answer: $x(t)$ is an antiderivative of $f(t)$. Thus to find all solutions, we just integrate both sides of (4.1.1) with respect to t. Formally, using indefinite integrals,

$$\int \frac{dx}{dt}(t)dt = \int f(t)dt + C, \tag{4.1.2}$$

where C is an arbitrary constant. (It is tempting to put a constant of integration on both sides of (4.1.2), but two constants are not needed, as we can just combine both constants on the right hand side of this equation.) Since the indefinite integral of dx/dt is just the function $x(t)$, we have

$$x(t) = \int f(\tau)d\tau + C. \tag{4.1.3}$$

In particular, finding closed form solutions to differential equations of the type (4.1.1) is equivalent to finding all definite integrals of the function $f(t)$. Indeed, to find closed form solutions to differential equations like (4.1.1) we need to know all of the techniques of integration from integral calculus.

We note that if $x(t)$ is a real-valued function of t, then we denote the derivative of x with respect to t using the following

$$\frac{dx}{dt} \qquad \dot{x} \qquad x'$$

all of which are standard notations for the derivative.

Initial Conditions and the Role of the Integration Constant C Equation (4.1.3) tells us that there are an infinite number of solutions to the differential equation (4.1.1), each one corresponding to a different choice of the constant C. To understand how to interpret the constant C, consider the example

$$\frac{dx}{dt}(t) = \cos t.$$

Using (4.1.3) we see that the answer is

$$x(t) = \int \cos\tau d\tau + C = \sin t + C.$$

Note that

$$x(0) = \sin(0) + C = C.$$

Thus, the constant C represents an *initial condition* for the differential equation. We will return to the discussion of initial conditions several times in this chapter.

The Linear Differential Equation of Growth and Decay

The subject of differential equations that we study begins when the function f on the right hand side of (4.1.1) depends explicitly on the function x, and the simplest such differential equation is:

$$\frac{dx}{dt}(t) = x(t).$$

Using results from differential calculus, we can solve this equation; indeed, we can solve the slightly more complicated equation

$$\frac{dx}{dt}(t) = \lambda x(t), \tag{4.1.4}$$

where $\lambda \in \mathbb{R}$ is a constant. The differential equation (4.1.4) is *linear* since $x(t)$ appears by itself on the right hand side. Moreover, (4.1.4) is *homogeneous* since the constant function $x(t) = 0$ is a solution.

In words (4.1.4) asks: For which functions $x(t)$ is the derivative of $x(t)$ equal to $\lambda x(t)$. The function

$$x(t) = e^{\lambda t}$$

is such a function, since

$$\frac{dx}{dt}(t) = \frac{d}{dt}e^{\lambda t} = \lambda e^{\lambda t} = \lambda x(t).$$

More generally, the function

$$x(t) = Ke^{\lambda t} \tag{4.1.5}$$

is a solution to (4.1.4) for any real constant K. We claim that the functions (4.1.5) list all (differentiable) functions that solve (4.1.4).

To verify this claim, we let $x(t)$ be a solution to (4.1.4) and show that the ratio

$$\frac{x(t)}{e^{\lambda t}} = x(t)e^{-\lambda t}$$

is a constant (independent of t). Using the product rule and (4.1.4), compute

$$\begin{aligned}
\frac{d}{dt}\left[x(t)e^{-\lambda t}\right] &= \frac{d}{dt}\left(x(t)\right)e^{-\lambda t} + x(t)\frac{d}{dt}\left(e^{-\lambda t}\right) \\
&= (\lambda x(t))e^{-\lambda t} + x(t)(-\lambda e^{-\lambda t}) \\
&= 0.
\end{aligned}$$

Now recall that the only functions whose derivatives are identically zero are the constant functions. Thus,

$$x(t)e^{-\lambda t} = K$$

for some constant $K \in \mathbb{R}$. Hence $x(t)$ has the form (4.1.5), as claimed.

Next, we discuss the role of the constant K. We have written the function as $x(t)$, and we have meant the reader to think of the variable t as time. Thus $x(0)$ is the initial value of the function $x(t)$ at time $t = 0$; we say that $x(0)$ is the *initial value* of $x(t)$. From (4.1.5) we see that

$$x(0) = K,$$

and that K is the initial value of the solution of (4.1.4). Henceforth, we write K as x_0 so that the notation calls attention to the special meaning of this constant.

By deriving (4.1.5) we have proved:

Theorem 4.1.1. *There is a unique solution to the* initial value problem

$$\begin{aligned}
\frac{dx}{dt}(t) &= \lambda x(t) \\
x(0) &= x_0.
\end{aligned} \tag{4.1.6}$$

That solution is

$$x(t) = x_0 e^{\lambda t}.$$

As a consequence of Theorem 4.1.1 we see that there is a qualitative difference in the behavior of solutions to

(4.1.6) depending on whether $\lambda > 0$ or $\lambda < 0$. Suppose that $x_0 > 0$. Then

$$\lim_{t\to\infty} x(t) = \lim_{t\to\infty} x_0 e^{\lambda t} = \begin{cases} +\infty & \lambda > 0 \\ 0 & \lambda < 0. \end{cases} \tag{4.1.7}$$

When $\lambda > 0$ we say that the solution has *exponential growth* and when $\lambda < 0$ we say that the solution has *exponential decay* . In either case, however, the number λ is called the *growth rate*. We can visualize this discussion by graphing the solutions in MATLAB.

Suppose we set $x_0 = 1$ and $\lambda = \pm 0.5$. Type

```
x0 = 1;
lambda = 0.5;
t = linspace(-1,4,100);
x = x0*exp(lambda*t);
plot(t,x)
hold on
xlabel('t')
ylabel('x')
lambda = -0.5;
x = x0*exp(lambda*t);
plot(t,x)
```

The result of this calculation is shown in Figure 12. In this way we can actually see the difference between exponential growth ($\lambda = 0.5$) and exponential decay ($\lambda = -0.5$), as discussed in the limit in (4.1.7).

Exercises

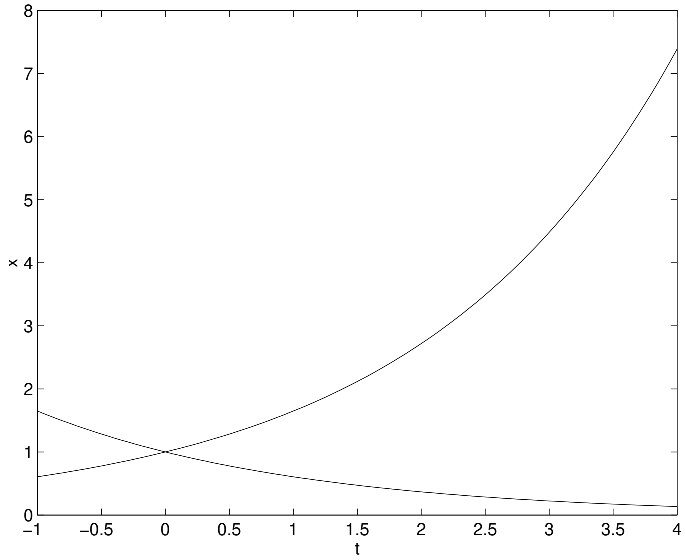

Figure 12: Solutions of (4.1.4) for $t \in [-1, 4]$, $x_0 = 1$ and $\lambda = \pm 0.5$.

4.2 *Rate Problems

Even though the homogeneous linear differential equation (4.1.6) is one of the simplest differential equations, it still has some use in applications. We present two here: compound interest and population dynamics.

Compound Interest Banks pay interest on an account in the following way. At the end of each day, the bank determines the interest rate r_{day} for that day, checks the principal P in the account, and then deposits an additional $r_{day}P$. So the next day the principal in this account is $(1 + r_{day})P$. Note that if r denotes the interest rate per year, then $r_{day} = r/365$. Of course, a day is just a convenient measure for elapsed time. Before computers were prevalent, banks paid interest yearly or quarterly or monthly or, in a few cases, even weekly, depending on the particular bank rules.

Observe that the more frequently interest is paid, the more money is earned. For example, if interest is paid only once at the end of a year, then the money in the account at the end of the year is $(1+r)P$, and the amount rP is called *simple interest*. But if interest is paid twice a year, then the principal at the end of six months will be $(1+\frac{r}{2})P$, and the principal at the end of the year will be $(1+\frac{r}{2})^2P$. Since

$$\left(1+\frac{r}{2}\right)^2 = 1+r+\frac{1}{4}r^2 > 1+r,$$

there is more money in the account at the end of the year if the interest is compounded semiannually rather than annually. But how much is the difference and what is the maximum earning potential?

While making the calculation in the previous paragraph, we implicitly made a number of simplifying assumptions. In particular, we assumed

- an initial principal P_0 is deposited in the bank on January 1,
- the money is not withdrawn for one year,
- no new money is deposited in that account during the year,
- the yearly interest rate r remains constant throughout the year, and
- interest is added to the account N times during the year.

In this *model*, simple interest corresponds to $N = 1$, compound monthly interest to $N = 12$, and compound daily interest to $N = 365$.

We first answer the question: How much money is in this account after one year? After one time unit of $\frac{1}{N}$ year, the amount of money in the account is

$$Q_1 = \left(1+\frac{r}{N}\right)P_0.$$

The interest rate in each time period is $\frac{r}{N}$, the yearly rate r divided by the number of time periods N. Here we have used the assumption that the interest rate remains constant throughout the year. After two time units, the principal is:

$$Q_2 = \left(1+\frac{r}{N}\right)Q_1 = \left(1+\frac{r}{N}\right)^2 P_0,$$

and at the end of the year (that is, after N time periods)

$$Q_N = \left(1+\frac{r}{N}\right)^N P_0. \tag{4.2.1}$$

Here we have used the assumption that money is neither deposited nor withdrawn from our account. Note that Q_N is the amount of money in the bank after **one** year assuming that interest has been compounded N (equally spaced) times during that year, and the effective interest rate when compounding N times is:

$$\left(1+\frac{r}{N}\right)^N - 1.$$

For the curious, we can write a program in MATLAB to compute (4.2.1). Suppose we assume that the initial deposit $P_0 = \$1,000$, the simple interest rate is 6% per year, and the interest payments are made monthly. In MATLAB type

```
N  = 12;
P0 = 1000;
r  = 0.06;
QN = (1 + r/N)^N*P0
```

The answer is $QN = \$1,061.68$, and the *effective* interest rate for monthly payments is 6.16778%. For daily interest payments $N = 365$, the answer is $QN = \$1,061.83$, and the effective interest rate is 6.18313%.

To find the maximum effective interest, we ask the bank to compound interest continuously; that is, we ask the bank to compute

$$\lim_{N\to\infty}\left(1+\frac{r}{N}\right)^N.$$

We compute this limit using differential equations. The concept of continuous interest is rephrased as follows. Let $P(t)$ be the principal at time t, where t is measured in units of years. Suppose that we assume that interest is compounded N times during the year. The length of time in each compounding period is

$$\Delta t = \frac{1}{N},$$

and the change in principal during that time period is

$$\Delta P = \frac{r}{N}P = rP\Delta t.$$

It follows that

$$\frac{\Delta P}{\Delta t} = rP,$$

and, on taking the limit $\Delta t \to 0$, we have the differential equation

$$\frac{dP}{dt}(t) = rP(t).$$

Since $P(0) = P_0$ the solution of the initial value problem given in Theorem 4.1.1 shows that

$$P(t) = P_0 e^{rt}.$$

After one year ($t = 1$) we find that

$$P(1) = e^r P_0.$$

Note that

$$P(1) = \lim_{N\to\infty} Q_N,$$

and we have thus verified that

$$\lim_{N\to\infty}\left(1+\frac{r}{N}\right)^N = e^r.$$

Thus the maximum effective interest rate is $e^r - 1$. When $r = 6\%$ the maximum effective interest rate is 6.18365%.

Population Dynamics To provide a second interpretation of the constant λ in (4.1.4), we discuss a simplified model for population dynamics. Let $p(t)$ be the size of a population of a certain species at time t and let r be the rate at which the population p is changing at time t. In general, r depends on the time t and is a complicated function of birth and death rates and of immigration and emigration, as well as of other factors. Indeed, the rate r may well depend on the size of the population itself. (Overcrowding can be modeled by assuming that the death rate increases with the size of the population.) These population models assume that the rate of change in the size of the population dp/dt is given by

$$\frac{dp}{dt}(t) = rp(t), \tag{4.2.2}$$

they just differ on the precise form of r. In general, the rate r will depend on the size of the population p as well as the time t, that is, r is a function $r(p, t)$.

The simplest population model — which we now assume — is the one in which r is assumed to be constant. Then equation (4.2.2) is identical to (4.1.4) after identifying p with x and r with λ. Hence we may interpret r as the growth rate for the population. The form of the solution in (4.1.5) shows that the size of a population grows exponentially if $r > 0$ and decays exponentially if $r < 0$.

The mathematical description of this simplest population model shows that the assumption of a constant growth rate leads to exponential growth (or exponential decay). Is this realistic? Surely, no population will grow exponentially for all time, and other factors, such as limited living space, have to be taken into account. On the other hand, exponential growth describes well the growth in human population during much of human history. So this model, though surely oversimplified, gives some insight into population growth.

Exercises

4.3 Uncoupled Linear Systems of Two Equations

A system of two linear ordinary differential equations has the form

$$\begin{aligned} \frac{dx}{dt}(t) &= ax(t)+by(t) \\ \frac{dy}{dt}(t) &= cx(t)+dy(t), \end{aligned} \tag{4.3.1}$$

where a,b,c,d are real constants. Solutions of (4.3.1) are pairs of functions $(x(t),y(t))$.

A solution to the planar system (4.3.1) that is constant in time t is called an *equilibrium*. Observe that the origin $(x(t),y(t))=(0,0)$ is always an equilibrium solution to the linear system (4.3.1).

We begin our discussion of linear systems of differential equations by considering uncoupled systems of the form

$$\begin{aligned} \frac{dx}{dt}(t) &= ax(t) \\ \frac{dy}{dt}(t) &= dy(t). \end{aligned} \tag{4.3.2}$$

Since the system is *uncoupled* (that is, the equation for $\dot{x}$ does not depend on y and the equation for $\dot{y}$ does not depend on x), we can solve this system by solving each equation independently, as we did for (4.1.4):

$$\begin{aligned} x(t) &= x_0e^{at} \\ y(t) &= y_0e^{dt}. \end{aligned} \tag{4.3.3}$$

There are now two initial conditions that are identified by

$$x(0)=x_0 \quad \text{and} \quad y(0)=y_0.$$

Having found all the solutions to (4.3.2) in (4.3.3), we now explore the geometry of the phase plane for these uncoupled systems both analytically and by using MATLAB.

Asymptotic Stability of the Origin As we did for the single equation (4.1.4), we ask what happens to solutions to (4.3.2) starting at (x_0,y_0) as time t increases. That is, we compute

$$\lim_{t\to\infty}(x(t),y(t)) = \lim_{t\to\infty}(x_0e^{at},y_0e^{dt}).$$

This limit is $(0,0)$ when both $a<0$ and $d<0$; but if either a or d is positive, then most solutions diverge to infinity, since either

$$\lim_{t\to\infty}|x(t)|=\infty \quad \text{or} \quad \lim_{t\to\infty}|y(t)|=\infty.$$

Roughly speaking, an equilibrium (x_0,y_0) is *asymptotically stable* if every trajectory $(x(t),y(t))$ beginning from an initial condition near (x_0,y_0) stays near (x_0,y_0) for all positive t, and

$$\lim_{t\to\infty}(x(t),y(t)) = (x_0,y_0).$$

The equilibrium is *unstable* if there are trajectories with initial conditions arbitrarily close to the equilibrium that move far away from that equilibrium.

At this stage, it is not clear how to determine whether the origin is asymptotically stable for a general linear system (4.3.1). However, for uncoupled linear systems we have shown that the origin is an asymptotically stable equilibrium when both $a<0$ and $d<0$. If either $a>0$ or $d>0$, then $(0,0)$ is unstable.

Invariance of the Axes There is another observation that we can make for uncoupled systems. Suppose that the initial condition for an uncoupled system lies on the x-axis; that is, suppose $y_0=0$. Then the solution $(x(t),y(t))=(x_0e^{at},0)$ also lies on the x-axis for all time. Similarly, if the initial condition lies on the y-axis, then the solution $(0,y_0e^{dt})$ lies on the y-axis for all time.

This invariance of the coordinate axes for uncoupled systems follows directly from (4.3.3). It turns out that many linear systems of differential equations have invariant lines; this is a topic to which we return later in this chapter.

Generating Phase Space Pictures with pplane10 How can we visualize a solution $(x(t), y(t))$ in (4.3.3) to the system of differential equations (4.3.2)? The time series approach suggests that we should graph $(x(t), y(t))$ as a function of t; that is, we should plot the curve

$$(t, x(t), y(t))$$

in three dimensions. Using MATLAB it is possible to plot such a graph — but such a graph by itself is difficult to interpret. Alternatively, we could graph either of the functions $x(t)$ or $y(t)$ by themselves as we do for solutions to single equations — but then some information is lost.

The method we prefer is the *phase space* plot obtained by thinking of $(x(t), y(t))$ as the position of a particle in the xy-plane at time t. We then graph the point $(x(t), y(t))$ in the plane as t varies. When looking at phase space plots, it is natural to call solutions *trajectories*, since we can imagine that we are watching a particle moving in the plane as time changes.

We begin by considering uncoupled linear equations. As we saw, when the initial conditions are on a coordinate axis (either $(x_0, 0)$ or $(0, y_0)$), the solutions remain on that coordinate axis for all time t. For these initial conditions, the equations behave as if they were one dimensional. However, if we consider an initial condition (x_0, y_0) that is not on a coordinate axis, then even for an uncoupled system it is a little difficult to *see* what the trajectory looks like. At this point it is useful to use the computer.

The method used to integrate planar systems of differential equations is similar to that used to integrate single equations. The solution curve $(x(t), y(t))$ to (4.3.2) at a point (x_0, y_0) is tangent to the direction $(f, g) = (ax_0 + by_0, cx_0. + dy_0)$. So the differential equation solver plots the direction field (f, g) and then finds curves that are tangent to these vectors at each point in time.

The program pplane10, written by John Polking, draws two-dimensional phase planes. In MATLAB type

```
pplane10
```

and the window with the PPLANE Setup appears. pplane10 has a number of preprogrammed differential equations listed in a menu accessed by clicking on Gallery. To explore linear systems, choose linear system in the Gallery. (Note that the parameters in the linear system are given by capitals rather than lower case a,b,c,d.)

To integrate the uncoupled linear system, set the parameters b and c equal to zero. We now have the system (4.3.2) with $a = 2$ and $d = -3$. After pushing Proceed, a display window appears. In this window the plane is filled by vectors (f, g) indicating directions.

We may start the computations by clicking with a mouse button on an initial value (x_0, y_0). For example, if we click approximately onto $(x(0), y(0)) = (x_0, y_0) = (1, 1)$, then the trajectory in the upper right quadrant of Figure 13 displays.

First pplane10 draws the trajectory in forward time for $t \geq 0$ and then it draws the trajectory in backwards time for $t \leq 0$. More precisely, when we click on a point (x_0, y_0) in the (x, y)-plane, pplane10 computes that part of the solution that lies inside the specified display window and that goes through this point. For linear systems there is precisely one solution that goes through a specified point in the (x, y)-plane.

Saddles, Sinks, and Sources for the Uncoupled System (4.3.2) In a qualitative fashion, the trajectories of uncoupled linear systems are determined by the invariance of the coordinate axes and by the signs of the constants

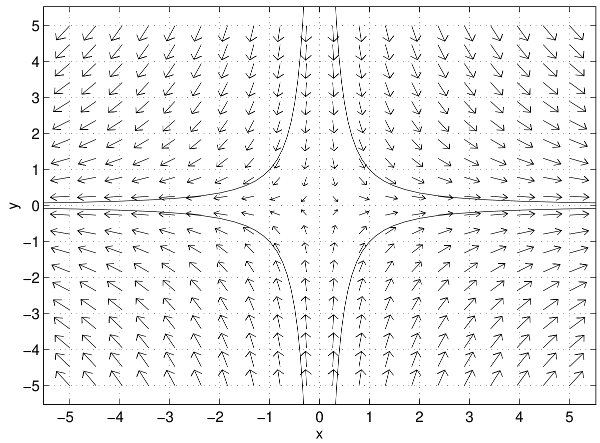

Figure 13: PPLANE10 Display for (4.3.2) with $a = 2$, $d = -3$ and $x, y \in [-5, 5]$. Solutions going through $(\pm 1, \pm 1)$ are shown.

a and d.

Saddles: $ad < 0$ In Figure 13, where $a = 2 > 0$ and $d = -3 < 0$, the origin is a *saddle*. If we choose several initial values (x_0, y_0) one after another, then we find that as time increases all solutions approach the x-axis. That is, if $(x(t), y(t))$ is a solution to this system of differential equations, then $\lim_{t\to\infty} y(t) = 0$. This observation is particularly noticeable when we choose initial conditions close to the origin $(0, 0)$. On the other hand, solutions also approach the y-axis as $t \to -\infty$. These qualitative features of the phase plane are valid whenever $a > 0$ and $d < 0$.

When $a < 0$ and $d > 0$, then the origin is also a saddle — but the roles of the x and y axes are reversed.

Sinks: $a < 0$ and $d < 0$ Now change the parameter a to -1. After clicking on Proceed and specifying several initial conditions, we see that all solutions approach the origin as time tends to infinity. Hence — as mentioned previously, and in contrast to saddles — the equilibrium $(0, 0)$ is asymptotically stable. Observe that solutions approach the origin on trajectories that are tangent to the x-axis. Since $d < a < 0$, the trajectory decreases to zero faster in the y direction than it does in the x-direction. If you change parameters so that $a < d < 0$, then trajectories will approach the origin tangent to the y-axis.

Sources: $a > 0$ and $d > 0$ Choose the constants a and d so that both are positive. In forward time, all trajectories, except the equilibrium at the origin, move towards infinity and the origin is called a *source*.

Time Series Using pplane10 We may also use pplane10 to graph the time series of the single components $x(t)$ and $y(t)$ of a solution $(x(t), y(t))$. For this we choose x vs. t from the Graph menu. After using the mouse to select a solution curve, another window with the title PPLANE t-plot appears. There the time series of $x(t)$ is shown. For example, when the differential equation is a sink, we observe that this component approaches 0 as time t tends to infinity. We may also display the time series of both components $x(t)$ and $y(t)$ simultaneously by clicking on Both in the PPLANE t-plot window. Again we see that both $x(t)$ and $y(t)$ tend to 0 for increasing t.

We may also visualize the time series of $x(t)$ and $y(t)$ in the three-dimensional (x, y, t)-space. To see this, click onto 3 D and a curve $(x(t), y(t), t)$ becomes visible. Since $x(t)$ and $y(t)$ approach 0 for $t \to \infty$ we see that this curve approaches the t-axis for increasing time t. Finally, we may look at all the different visualizations — the phase space plot, the time series for $x(t)$ and $y(t)$ and the three-dimensional representation of the solution — by clicking the Composite button. See Figure 14.

Exercises

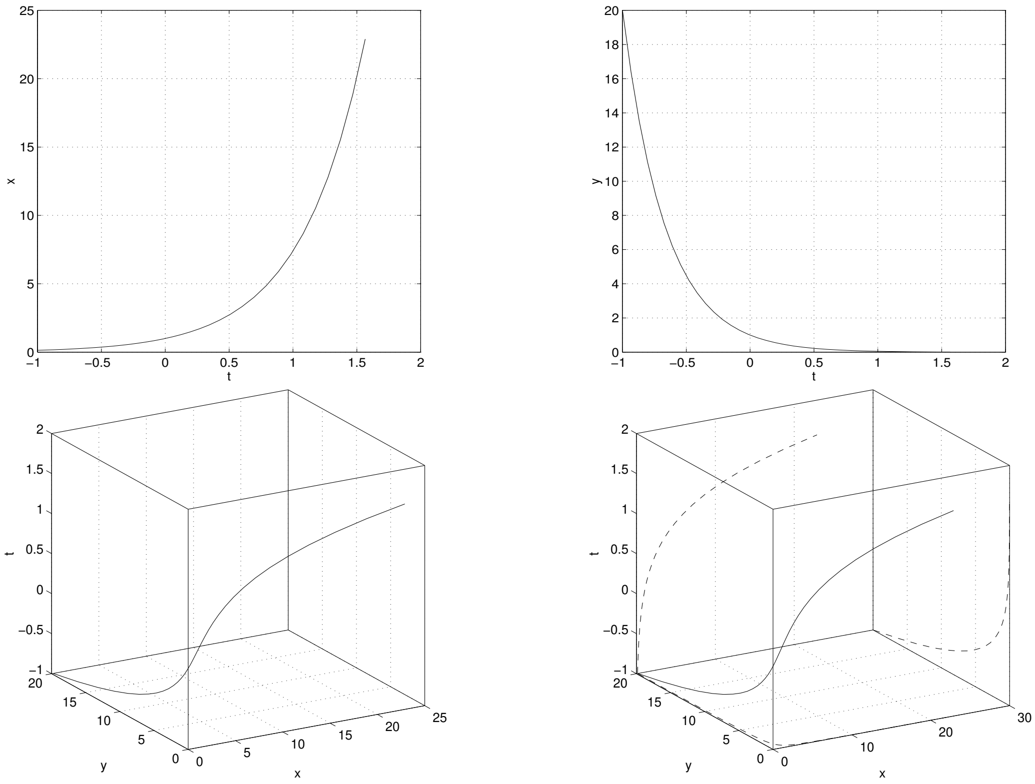

Figure 14: PPLANE9 Display for (4.3.2) with $a = 2$, $d = -3$ and $x \in [0, 25], y \in [0, 20]$. The solution going through $(1, 1)$ is shown. UL: $(t, x(t))$; UR: $(t, y(t))$; LL: $(x(t), y(t), t)$; LR: all plots.

4.4 Coupled Linear Systems

The general linear constant coefficient system in two unknown functions x_1, x_2 is:

$$\begin{aligned} \frac{dx_1}{dt}(t) &= ax_1(t) + bx_2(t) \\ \frac{dx_2}{dt}(t) &= cx_1(t) + dx_2(t). \end{aligned} \tag{4.4.1}$$

The uncoupled systems studied in Section 4.3 are obtained by setting $b = c = 0$ in (4.4.1). We have discussed how to solve (4.4.1) by formula (4.3.3) when the system is uncoupled. We have also discussed how to visualize the phase plane for different choices of the diagonal entries a and d. At present, we cannot solve (4.4.1) by formula when the coefficient matrix is not diagonal. But we may use pplane10 to solve the initial value problems numerically for these coupled systems. We illustrate this point by solving

$$\begin{aligned} \frac{dx_1}{dt}(t) &= -x_1(t) + 3x_2(t) \\ \frac{dx_2}{dt}(t) &= 3x_1(t) - x_2(t). \end{aligned}$$

After starting pplane10, select linear system from the Gallery and set the constants to:

$$a = -1, \quad b = 3, \quad c = 3, \quad d = -1.$$

Click on Proceed. In order to have equally spaced coordinates on the x and y axes, do the following. In the PPLANE9 Display window click on the edit button and then on the zoom in square command. Then, using the mouse, click on the origin.

Eigendirections After computing several solutions, we find that for increasing time t all the solutions seem to approach the diagonal line given by the equation $x_1 = x_2$. Similarly, in backward time t the solutions approach the anti-diagonal $x_1 = -x_2$. In other words, as for the case of uncoupled systems, we find two distinguished directions in the (x, y)-plane. See Figure 15. Moreover, the computations indicate that these lines are invariant in the sense that solutions starting on these lines remain on them for all time. This statement can be verified numerically by using the Keyboard input in the PPLANE10 Options to choose initial conditions $(x_0, y_0) = (1, 1)$ and $(x_0, y_0) = (1, -1)$.

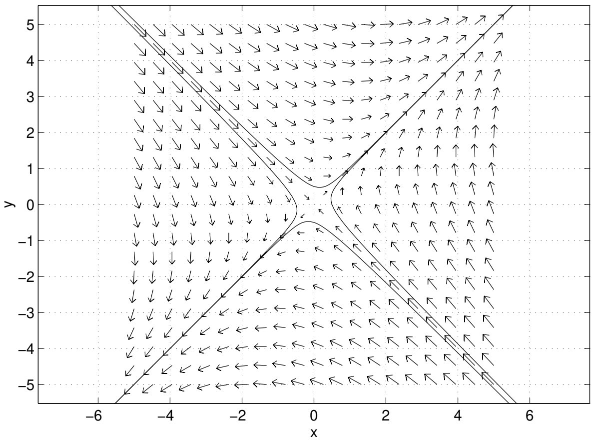

Figure 15: PPLANE10 Display for (4.4.1) with $a = -1 = d$; $b = 3 = c$; and $x, y \in [-5, 5]$. Solutions going through $(\pm 0.5, 0)$ and $(0, \pm 0.5)$ are shown.

Definition 4.4.1. An invariant line for a linear system of differential equations is called an *eigendirection.*

Observe that eigendirections vary if we change parameters. For example, if we set b to 1, then there are still two distinguished lines but these lines are no longer perpendicular.

For uncoupled systems, we have shown analytically that the x and y axes are eigendirections. The numerical computations that we have just performed indicate that eigendirections exist for many coupled systems. This discussion leads naturally to two questions:

(a) Do eigendirections always exist?

(b) How can we find eigendirections?

The second question will be answered in Sections 4.5 and 4.6. We can answer the first question by performing another numerical computation. In the setup window, change the parameter b to -2. Then numerically compute some solutions to see that there are no eigendirections in the phase space of this system. Observe that all solutions appear to spiral into the origin as time goes to infinity. The phase portrait is shown in Figure 16.

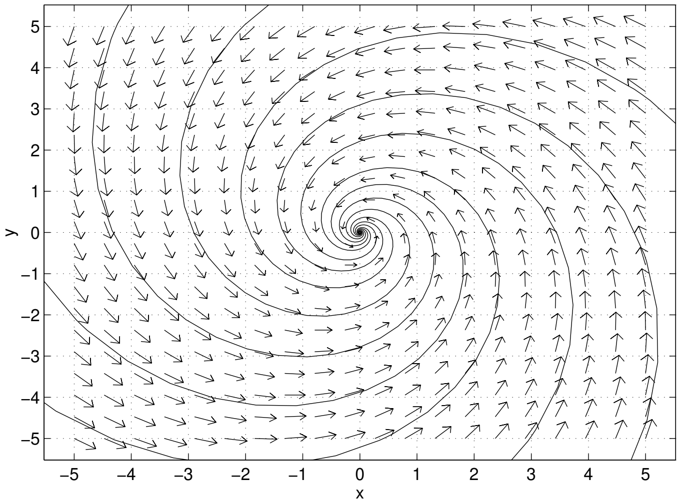

Figure 16: PPLANE10 Display for the linear system with $a = -1$, $b = -2$, $c = 3$, $d = -1$.

Nonexistence of Eigendirections We now show analytically that certain linear systems of differential equations have no invariant lines in their phase portrait. Consider the system

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -x. \end{aligned} \tag{4.4.2}$$

Observe that $(x(t), y(t)) = (\sin t, \cos t)$ is a solution to (4.4.2) by calculating

$$\begin{aligned} \dot{x}(t) &= \frac{d}{dt}\sin t &= \cos t &= y(t) \\ \dot{y}(t) &= \frac{d}{dt}\cos t &= -\sin t &= -x(t) \end{aligned}$$

We have shown analytically that the unit circle centered at the origin is a solution trajectory for (4.4.2). Hence (4.4.2) has no eigendirections. It may be checked using MATLAB that all solution trajectories for (4.4.2) are just circles centered at the origin.

Exercises

4.5 The Initial Value Problem and Eigenvectors

The general *constant coefficient* system of n differential equations in n unknown functions has the form

$$\begin{array}{rcl} \dfrac{dx_1}{dt}(t) & = & c_{11}x_1(t) + \cdots + c_{1n}x_n(t) \\ \vdots & \vdots & \vdots \\ \dfrac{dx_n}{dt}(t) & = & c_{n1}x_1(t) + \cdots + c_{nn}x_n(t) \end{array} \tag{4.5.1}$$

where the coefficients $c_{ij} \in \mathbb{R}$ are constants. Suppose that (4.5.1) satisfies the initial conditions

$$x_1(0) = K_1, \ldots, x_n(0) = K_n.$$

Using matrix multiplication of a vector and matrix, we can rewrite these differential equations in a compact form. Consider the $n \times n$ coefficient matrix

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

and the n vectors of initial conditions and unknowns

$$X_0 = \begin{pmatrix} K_1 \\ \vdots \\ K_n \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

Then (4.5.1) has the compact form

$$\begin{array}{rcl} \dfrac{dX}{dt} & = & CX \\ X(0) & = & X_0. \end{array} \tag{4.5.2}$$

In Section 4.4, we plotted the phase space picture of the planar system of differential equations

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = C \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \tag{4.5.3}$$

where

$$C = \begin{pmatrix} -1 & 3 \\ 3 & -1 \end{pmatrix}.$$

In those calculations we observed that there is a solution to (4.5.3) that stayed on the main diagonal for each moment in time. Note that a vector is on the main diagonal if it is a scalar multiple of $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Thus a solution that stays on the main diagonal for all time t must have the form

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = u(t) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{4.5.4}$$

for some real-valued function $u(t)$. When a function of form (4.5.4) is a solution to (4.5.3), it satisfies:

$$\begin{aligned} \dot{u}(t) \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} &= C \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \\ &= Cu(t) \begin{pmatrix} 1 \\ 1 \end{pmatrix} = u(t) C \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \end{aligned}$$

A calculation shows that

$$C \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Hence

$$\dot{u}(t) \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2u(t) \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

It follows that the function $u(t)$ must satisfy the differential equation

$$\frac{du}{dt} = 2u.$$

whose solutions are

$$u(t) = \alpha e^{2t},$$

for some scalar α.

Similarly, we also saw in our MATLAB experiments that there was a solution that for all time stayed on the antidiagonal, the line $y = -x$. Such a solution must have the

form
$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = v(t)\begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$
A similar calculation shows that $v(t)$ must satisfy the differential equation
$$\frac{dv}{dt} = -4v.$$
Solutions to this equation all have the form
$$v(t) = \beta e^{-4t},$$
for some real constant β.

Thus, using matrix multiplication, we are able to prove analytically that there are solutions to (4.5.3) of exactly the type suggested by our MATLAB experiments. However, even more is true and this extension is based on the principle of superposition that was introduced for algebraic equations in Section 3.4.

Superposition in Linear Differential Equations Consider a general linear differential equation of the form
$$\frac{dX}{dt} = CX, \tag{4.5.5}$$
where C is an $n \times n$ matrix. Suppose that $Y(t)$ and $Z(t)$ are solutions to (4.5.5) and $\alpha, \beta \in \mathbb{R}$ are scalars. Then $X(t) = \alpha Y(t) + \beta Z(t)$ is also a solution. We verify this fact using the 'linearity' of d/dt. Calculate
$$\begin{aligned} \frac{d}{dt}X(t) &= \alpha\frac{dY}{dt}(t) + \beta\frac{dZ}{dt}(t) \\ &= \alpha CY(t) + \beta CZ(t) \\ &= C(\alpha Y(t) + \beta Z(t)) \\ &= CX(t). \end{aligned}$$
So superposition is valid for solutions of linear differential equations.

Initial Value Problems Suppose that we wish to find a solution to (4.5.3) satisfying the initial conditions
$$\begin{pmatrix} x(0) \\ y(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}.$$
Then we can use the principle of superposition to find this solution in closed form. Superposition implies that for each pair of scalars $\alpha, \beta \in \mathbb{R}$, the functions
$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \alpha e^{2t}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \beta e^{-4t}\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \tag{4.5.6}$$
are solutions to (4.5.3). Moreover, for a solution of this form
$$\begin{pmatrix} x(0) \\ y(0) \end{pmatrix} = \begin{pmatrix} \alpha+\beta \\ \alpha-\beta \end{pmatrix}.$$
Thus we can solve our prescribed initial value problem, if we can solve the system of linear equations
$$\begin{aligned} \alpha + \beta &= 1 \\ \alpha - \beta &= 3. \end{aligned}$$
This system is solved for $\alpha = 2$ and $\beta = -1$. Thus
$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = 2e^{2t}\begin{pmatrix} 1 \\ 1 \end{pmatrix} - e^{-4t}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$
is the desired closed form solution.

Eigenvectors and Eigenvalues We emphasize that just knowing that there are two lines in the plane that are invariant under the dynamics of the system of linear differential equations is sufficient information to solve these equations. So it seems appropriate to ask the question: When is there a line that is invariant under the dynamics of a system of linear differential equations? This question is equivalent to asking: When is there a nonzero vector v and a nonzero real-valued function $u(t)$ such that
$$X(t) = u(t)v$$

is a solution to (4.5.5)?

Suppose that $X(t)$ is a solution to the system of differential equations $\dot{X} = CX$. Then $u(t)$ and v must satisfy

$$\dot{u}(t)v = \frac{dX}{dt} = CX(t) = u(t)Cv. \tag{4.5.7}$$

Since u is nonzero, it follows that v and Cv must lie on the same line through the origin. Hence

$$Cv = \lambda v, \tag{4.5.8}$$

for some real number λ.

Definition 4.5.1. A nonzero vector v satisfying (4.5.8) is called an *eigenvector* of the matrix C, and the number λ is an *eigenvalue* of the matrix C.

Geometrically, the matrix C maps an eigenvector onto a multiple of itself — that multiple is the eigenvalue.

Note that scalar multiples of eigenvectors are also eigenvectors. More precisely:

Lemma 4.5.2. *Let v be an eigenvector of the matrix C with eigenvalue λ. Then αv is also an eigenvector of C with eigenvalue λ as long as $\alpha \neq 0$.*

Proof By assumption, $Cv = \lambda v$ and v is nonzero. Now calculate

$$C(\alpha v) = \alpha Cv = \alpha\lambda v = \lambda(\alpha v).$$

The lemma follows from the definition of eigenvector. ■

It follows from (4.5.7) and (4.5.8) that if v is an eigenvector of C with eigenvalue λ, then

$$\frac{du}{dt} = \lambda u.$$

Thus we have returned to our original linear differential equation that has solutions

$$u(t) = Ke^{\lambda t},$$

for all constants K.

We have proved the following theorem.

Theorem 4.5.3. *Let v be an eigenvector of the $n \times n$ matrix C with eigenvalue λ. Then*

$$X(t) = e^{\lambda t}v$$

is a solution to the system of differential equations $\dot{X} = CX$.

Finding eigenvalues and eigenvectors from first principles — even for 2×2 matrices — is not a simple task. We end this section with a calculation illustrating that real eigenvalues need not exist. In Section 4.6, we present a natural method for computing eigenvalues (and eigenvectors) of 2×2 matrices. We defer the discuss of how to find eigenvalues and eigenvectors of $n \times n$ matrices until Chapter 7.

An Example of a Matrix with No Real Eigenvalues

Not every matrix has *real* eigenvalues and eigenvectors. Recall the linear system of differential equations $\dot{x} = Cx$ whose phase plane is pictured in Figure 16. That phase plane showed no evidence of an invariant line and indeed there is none. The matrix C in that example was

$$C = \begin{pmatrix} -1 & -2 \\ 3 & -1 \end{pmatrix}.$$

We ask: Is there a value of λ and a nonzero vector (x, y) such that

$$C\begin{pmatrix} x \\ y \end{pmatrix} = \lambda\begin{pmatrix} x \\ y \end{pmatrix}? \tag{4.5.9}$$

Equation (4.5.9) implies that

$$\begin{pmatrix} -1-\lambda & -2 \\ 3 & -1-\lambda \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = 0.$$

If this matrix is row equivalent to the identity matrix, then the only solution of the linear system is $x = y = 0$. To have a nonzero solution, the matrix

$$\begin{pmatrix} -1-\lambda & -2 \\ 3 & -1-\lambda \end{pmatrix}$$

must not be row equivalent to I_2. Dividing the 1^{st} row by $-(1+\lambda)$ leads to

$$\begin{pmatrix} 1 & \dfrac{2}{1+\lambda} \\ 3 & -1-\lambda \end{pmatrix}.$$

Subtracting 3 times the 1^{st} row from the second produces the matrix

$$\begin{pmatrix} 1 & \dfrac{2}{1+\lambda} \\ 0 & -(1+\lambda) - \dfrac{6}{1+\lambda} \end{pmatrix}.$$

This matrix is not row equivalent to I_2 when the lower right hand entry is zero; that is, when

$$(1+\lambda) + \frac{6}{1+\lambda} = 0.$$

That is, when

$$(1+\lambda)^2 = -6,$$

which is not possible for any real number λ. This example shows that the question of whether a given matrix has a real eigenvalue and a real eigenvector — and hence when the associated system of differential equations has a line that is invariant under the dynamics — is a subtle question.

Questions concerning eigenvectors and eigenvalues are central to much of the theory of linear algebra. We discuss this topic for 2×2 matrices in Section 4.6 and Chapter 6 and for general square matrices in Chapters 7 and 11.

Exercises

4.6 Eigenvalues of 2×2 Matrices

We now discuss how to find eigenvalues of 2×2 matrices in a way that does not depend explicitly on finding eigenvectors. This direct method will show that eigenvalues can be complex as well as real.

We begin the discussion with a general square matrix. Let A be an $n \times n$ matrix. Recall that $\lambda \in \mathbb{R}$ is an eigenvalue of A if there is a nonzero vector $v \in \mathbb{R}^n$ for which

$$Av = \lambda v. \tag{4.6.1}$$

The vector v is called an eigenvector. We may rewrite (4.6.1) as:

$$(A - \lambda I_n)v = 0.$$

Since v is nonzero, it follows that if λ is an eigenvalue of A, then the matrix $A - \lambda I_n$ is singular.

Conversely, suppose that $A - \lambda I_n$ is singular for some real number λ. Then Theorem 3.7.8 of Chapter 3 implies that there is a nonzero vector $v \in \mathbb{R}^n$ such that $(A - \lambda I_n)v = 0$. Hence (4.6.1) holds and λ is an eigenvalue of A. So, if we had a direct method for determining when a matrix is singular, then we would have a method for determining eigenvalues.

Characteristic Polynomials Corollary 3.8.3 of Chapter 3 states that 2×2 matrices are singular precisely when their determinant is zero. It follows that $\lambda \in \mathbb{R}$ is an eigenvalue for the 2×2 matrix A precisely when

$$\det(A - \lambda I_2) = 0. \tag{4.6.2}$$

We can compute (4.6.2) explicitly as follows. Note that

$$A - \lambda I_2 = \begin{pmatrix} a - \lambda & b \\ c & d - \lambda \end{pmatrix}.$$

Therefore

$$\begin{aligned} \det(A - \lambda I_2) &= (a - \lambda)(d - \lambda) - bc \\ &= \lambda^2 - (a + d)\lambda + (ad - bc) \end{aligned} \tag{4.6.3}$$

Definition 4.6.1. The *characteristic polynomial* of the matrix A is

$$p_A(\lambda) = \det(A - \lambda I_2).$$

For an $n \times n$ matrix $A = (a_{ij})$, define the *trace* of A to be the sum of the diagonal elements of A; that is

$$\mathrm{tr}(A) = a_{11} + \cdots + a_{nn}. \tag{4.6.4}$$

Thus, using (4.6.3), we can rewrite the characteristic polynomial for 2×2 matrices as

$$p_A(\lambda) = \lambda^2 - \mathrm{tr}(A)\lambda + \det(A). \tag{4.6.5}$$

As an example, consider the 2×2 matrix

$$A = \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix}. \tag{4.6.6}$$

Then

$$A - \lambda I_2 = \begin{pmatrix} 2 - \lambda & 3 \\ 1 & 4 - \lambda \end{pmatrix},$$

and

$$p_A(\lambda) = (2 - \lambda)(4 - \lambda) - 3 = \lambda^2 - 6\lambda + 5.$$

It is now easy to verify (4.6.5) for (4.6.6).

Eigenvalues For 2×2 matrices A, $p_A(\lambda)$ is a quadratic polynomial. As we have discussed, the real roots of p_A are real eigenvalues of A. For 2×2 matrices we now generalize our first definition of eigenvalues, Definition 4.5.1, to include complex eigenvalues, as follows.

Definition 4.6.2. An *eigenvalue* of A is a root of the characteristic polynomial p_A.

It follows from Definition 4.6.2 that every 2×2 matrix has precisely two eigenvalues, which may be equal or complex conjugate pairs.

Suppose that λ_1 and λ_2 are the roots of p_A. It follows that

$$p_A(\lambda) = (\lambda-\lambda_1)(\lambda-\lambda_2) = \lambda^2-(\lambda_1+\lambda_2)\lambda+\lambda_1\lambda_2. \quad (4.6.7)$$

Equating the two forms of p_A (4.6.5) and (4.6.7) shows that

$$\begin{aligned} \operatorname{tr}(A) &= \lambda_1 + \lambda_2 && (4.6.8) \\ \det(A) &= \lambda_1\lambda_2. && (4.6.9) \end{aligned}$$

Thus, for 2×2 matrices, the trace is the sum of the eigenvalues and the determinant is the product of the eigenvalues. In Chapter 7, Theorems 7.2.4(b) and 7.2.9 we show that these statements are also valid for $n \times n$ matrices.

Recall that in example (4.6.6) the characteristic polynomial is

$$p_A(\lambda) = \lambda^2 - 6\lambda + 5 = (\lambda - 5)(\lambda - 1).$$

Thus the eigenvalues of A are $\lambda_1 = 1$ and $\lambda_2 = 5$ and identities (4.6.8) and (4.6.9) are easily verified for this example.

Next, we consider an example with complex eigenvalues and verify that these identities are equally valid in this instance. Let

$$B = \begin{pmatrix} 2 & -3 \\ 1 & 4 \end{pmatrix}.$$

The characteristic polynomial is:

$$p_B(\lambda) = \lambda^2 - 6\lambda + 11.$$

Using the quadratic formula we see that the roots of p_B (that is, the eigenvalues of B) are

$$\lambda_1 = 3 + i\sqrt{2} \quad \text{and} \quad \lambda_2 = 3 - i\sqrt{2}.$$

Again the sum of the eigenvalues is 6 which equals the trace of B and the product of the eigenvalues is 11 which equals the determinant of B.

Since the characteristic polynomial of 2×2 matrices is always a quadratic polynomial, it follows that 2×2 matrices have precisely two eigenvalues — including multiplicity — and these can be described as follows. The *discriminant* of A is:

$$D = [\operatorname{tr}(A)]^2 - 4\det(A). \quad (4.6.10)$$

Theorem 4.6.3. *There are three possibilities for the two eigenvalues of a* 2×2 *matrix* A *that we can describe in terms of the discriminant:*

(i) The eigenvalues of A *are real and distinct* ($D > 0$)*.*

(ii) The eigenvalues of A *are a complex conjugate pair* ($D < 0$)*.*

(iii) The eigenvalues of A *are real and equal* ($D = 0$)*.*

Proof We can find the roots of the characteristic polynomial using the form of p_A given in (4.6.5) and the quadratic formula. The roots are:

$$\frac{1}{2}\left(\operatorname{tr}(A) \pm \sqrt{[\operatorname{tr}(A)]^2 - 4\det(A)}\right) = \frac{\operatorname{tr}(A) \pm \sqrt{D}}{2}.$$

The proof of the theorem now follows. If $D > 0$, then the eigenvalues of A are real and distinct; if $D < 0$, then eigenvalues are complex conjugates; and if $D = 0$, then the eigenvalues are real and equal. ■

Eigenvectors The following lemma contains an important observation about eigenvectors:

Lemma 4.6.4. *Every eigenvalue* λ *of a* 2×2 *matrix* A *has an eigenvector* v*. That is, there is a nonzero vector* $v \in \mathbb{C}^2$ *satisfying*

$$Av = \lambda v.$$

Proof When the eigenvalue λ is real we know that an eigenvector $v \in \mathbb{R}^2$ exists. However, when λ is complex, then we must show that there is a complex eigenvector $v \in \mathbb{C}^2$, and this we have not yet done. More precisely, we must show that if λ is a complex root of the characteristic polynomial p_A, then there is a complex vector v such that

$$(A - \lambda I_2)v = 0.$$

As we discussed in Section 2.5, finding v is equivalent to showing that the complex matrix

$$A - \lambda I_2 = \begin{pmatrix} a-\lambda & b \\ c & d-\lambda \end{pmatrix}$$

is not row equivalent to the identity matrix. See Theorem 2.5.2 of Chapter 2. Since a is real and λ is not, $a - \lambda \neq 0$. A short calculation shows that $A - \lambda I_2$ is row equivalent to the matrix

$$\begin{pmatrix} 1 & \dfrac{b}{a-\lambda} \\ 0 & \dfrac{p_A(\lambda)}{a-\lambda} \end{pmatrix}.$$

This matrix is not row equivalent to the identity matrix since $p_A(\lambda) = 0$. ■

An Example of a Matrix with Real Eigenvectors Once we know the eigenvalues of a 2×2 matrix, the associated eigenvectors can be found by direct calculation. For example, we showed previously that the matrix

$$A = \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix}.$$

in (4.6.6) has eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 5$. With this information we can find the associated eigenvectors. To find an eigenvector associated with the eigenvalue $\lambda_1 = 1$ compute

$$A - \lambda_1 I_2 = A - I_2 = \begin{pmatrix} 1 & 3 \\ 1 & 3 \end{pmatrix}.$$

It follows that $v_1 = (3, -1)^t$ is an eigenvector since

$$(A - I_2)v_1 = 0.$$

Similarly, to find an eigenvector associated with the eigenvalue $\lambda_2 = 5$ compute

$$A - \lambda_2 I_2 = A - 5I_2 = \begin{pmatrix} -3 & 3 \\ 1 & -1 \end{pmatrix}.$$

It follows that $v_2 = (1, 1)^t$ is an eigenvector since

$$(A - 5I_2)v_2 = 0.$$

Examples of Matrices with Complex Eigenvectors Let

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

Then $p_A(\lambda) = \lambda^2 + 1$ and the eigenvalues of A are $\pm i$. To find the eigenvector $v \in \mathbb{C}^2$ whose existence is guaranteed by Lemma 4.6.4, we need to solve the complex system of linear equations $Av = iv$. We can rewrite this system as:

$$\begin{pmatrix} -i & -1 \\ 1 & -i \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0.$$

A calculation shows that

$$v = \begin{pmatrix} i \\ 1 \end{pmatrix} \tag{4.6.11}$$

is a solution. Since the coefficients of A are real, we can take the complex conjugate of the equation $Av = iv$ to obtain

$$A\overline{v} = -i\overline{v}.$$

Thus

$$\overline{v} = \begin{pmatrix} -i \\ 1 \end{pmatrix}$$

is the eigenvector corresponding to the eigenvalue $-i$. This comment is valid for any complex eigenvalue.

More generally, let

$$A = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix}, \tag{4.6.12}$$

where $\tau \neq 0$. Then

$$\begin{aligned} p_A(\lambda) &= \lambda^2 - 2\sigma\lambda + \sigma^2 + \tau^2 \\ &= (\lambda - (\sigma + i\tau))(\lambda - (\sigma - i\tau)), \end{aligned}$$

and the eigenvalues of A are the complex conjugates $\sigma \pm i\tau$. Thus A has no real eigenvectors. The complex eigenvectors of A are v and $\overline{v}$ where v is defined in (4.6.11).

Exercises

4.7 Initial Value Problems Revisited

To summarize the ideas developed in this chapter, we review the method that we have developed to solve the system of differential equations

$$\begin{array}{rcl} \dot{x} & = & ax+by \\ \dot{y} & = & cx+dy \end{array} \tag{4.7.1}$$

satisfying the initial conditions

$$\begin{array}{rcl} x(0) & = & x_0 \\ y(0) & = & y_0. \end{array} \tag{4.7.2}$$

Begin by rewriting (4.7.1) in matrix form

$$\dot{X} = CX \tag{4.7.3}$$

where

$$C = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{and} \quad X(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}.$$

Rewrite the initial conditions (4.7.2) in vector form

$$X(0) = X_0 \tag{4.7.4}$$

where

$$X_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}.$$

When the eigenvalues of C are *real* and *distinct* we now know how to solve the initial value problem (4.7.3) and (4.7.4). This solution is found in four steps.

Step 1: Find the eigenvalues λ_1 and λ_2 of C.

These eigenvalues are the roots of the characteristic polynomial as given by (4.6.5):

$$p_C(\lambda) = \lambda^2 - \mathrm{tr}(C)\lambda + \det(C).$$

These roots may be found either by factoring p_C or by using the quadratic formula. The roots are real and distinct when the discriminant

$$D = \mathrm{tr}(C)^2 - 4\det(C) > 0.$$

Recall (4.6.10) and Theorem 4.6.3.

Step 2: Find eigenvectors v_1 and v_2 of C associated with the eigenvalues λ_1 and λ_2.

For $j = 1$ and $j = 2$, the eigenvector v_j is found by solving the homogeneous system of linear equations

$$(C - \lambda_j I_2)v = 0 \tag{4.7.5}$$

for one nonzero solution. Lemma 4.6.4 tells us that there is always a nonzero solution to (4.7.5) since λ_j is an eigenvalue of C.

Step 3: Using superposition, write the *general solution* to the system of ODEs (4.7.3) as

$$X(t) = \alpha_1 e^{\lambda_1 t} v_1 + \alpha_2 e^{\lambda_2 t} v_2, \tag{4.7.6}$$

where $\alpha_1, \alpha_2 \in \mathbb{R}$.

Theorem 4.5.3 tells us that for $j = 1, 2$

$$X_j(t) = e^{\lambda_j t} v_j$$

is a solution to (4.7.3). The principle of superposition (see Section 4.5) allows us to conclude that

$$X(t) = \alpha_1 X_1(t) + \alpha_2 X_2(t)$$

is also a solution to (4.7.3) for any scalars $\alpha_1, \alpha_2 \in \mathbb{R}$. Thus, (4.7.6) is valid.

Note that the initial condition corresponding to the general solution (4.7.6) is

$$X(0) = \alpha_1 v_1 + \alpha_2 v_2, \tag{4.7.7}$$

since $e^0 = 1$.

Step 4: Solve the initial value problem by solving the system of linear equations

$$X_0 = \alpha_1 v_1 + \alpha_2 v_2 \tag{4.7.8}$$

for α_1 and α_2 (see (4.7.7)).

Let A be the 2×2 matrix whose columns are v_1 and v_2. That is,

$$A = (v_1|v_2). \tag{4.7.9}$$

Then we may rewrite (4.7.8) in the form

$$A \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = X_0. \tag{4.7.10}$$

We claim that the matrix $A = (v_1|v_2)$ (defined in (4.7.9)) is always invertible. Recall Lemma 4.5.2 which states that if w is a nonzero multiple of v_2, then w is also an eigenvector of A associated to the eigenvalue λ_2. Since the eigenvalues λ_1 and λ_2 are distinct, it follows that the eigenvector v_1 is not a scalar multiple of the eigenvector v_2 (see Lemma 4.5.2). Therefore, the area of the parallelogram spanned by v_1 and v_2 is nonzero and the determinant of A is nonzero by Theorem 3.8.5 of Chapter 3. Corollary 3.8.3 of Chapter 3 now implies that A is invertible. Thus, the unique solution to (4.7.10) is

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = A^{-1} X_0.$$

This equation is easily solved since we have an explicit formula for A^{-1} when A is a 2×2 matrix (see (3.8.1) in Section 3.8). Indeed,

$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

An Initial Value Problem Solved by Hand Solve the linear system of differential equations

$$\begin{aligned} \dot{x} &= 3x - y \\ \dot{y} &= 4x - 2y \end{aligned} \tag{4.7.11}$$

with initial conditions

$$\begin{aligned} x(0) &= 2 \\ y(0) &= -3. \end{aligned} \tag{4.7.12}$$

Rewrite the system (4.7.11) in matrix form as

$$\dot{X} = CX$$

where

$$C = \begin{pmatrix} 3 & -1 \\ 4 & -2 \end{pmatrix}.$$

Rewrite the initial conditions (4.7.12) in vector form

$$X(0) = X_0 = \begin{pmatrix} 2 \\ -3 \end{pmatrix}.$$

Now proceed through the four steps outlined previously.

Step 1: Find the eigenvalues of C.

The characteristic polynomial of C is

$$p_C(\lambda) = \lambda^2 - \mathrm{tr}(C)\lambda + \det(C) = \lambda^2 - \lambda - 2 = (\lambda - 2)(\lambda + 1).$$

Therefore, the eigenvalues of C are

$$\lambda_1 = 2 \quad \text{and} \quad \lambda_2 = -1.$$

Step 2: Find the eigenvectors of C.

Find an eigenvector associated with the eigenvalue $\lambda_1 = 2$ by solving the system of equations

$$\begin{aligned} (C - \lambda_1 I_2)v &= \left(\begin{pmatrix} 3 & -1 \\ 4 & -2 \end{pmatrix} - \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \right) v \\ &= \begin{pmatrix} 1 & -1 \\ 4 & -4 \end{pmatrix} v = 0. \end{aligned}$$

One particular solution to this system is

$$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Similarly, find an eigenvector associated with the eigenvalue $\lambda_2 = -1$ by solving the system of equations

$$\begin{aligned}(C - \lambda_2 I_2)v &= \left(\begin{pmatrix} 3 & -1 \\ 4 & -2 \end{pmatrix} - \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}\right) v \\ &= \begin{pmatrix} 4 & -1 \\ 4 & -1 \end{pmatrix} v = 0.\end{aligned}$$

One particular solution to this system is

$$v_2 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}.$$

Step 3: Write the general solution to the system of differential equations.

Using superposition the general solution to the system (4.7.11) is:

$$X(t) = \alpha_1 e^{2t} v_1 + \alpha_2 e^{-t} v_2 = \alpha_1 e^{2t} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \alpha_2 e^{-t} \begin{pmatrix} 1 \\ 4 \end{pmatrix},$$

where $\alpha_1, \alpha_2 \in \mathbb{R}$. Note that the initial state of this solution is:

$$X(0) = \alpha_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} \alpha_1 + \alpha_2 \\ \alpha_1 + 4\alpha_2 \end{pmatrix}.$$

Step 4: Solve the initial value problem.

Let

$$A = (v_1 | v_2) = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}.$$

The equation for the initial condition is

$$A \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = X_0.$$

See (4.7.9).

We can write the inverse of A by formula as

$$A^{-1} = \frac{1}{3} \begin{pmatrix} 4 & -1 \\ -1 & 1 \end{pmatrix}.$$

It follows that we solve for the coefficients α_j as

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = A^{-1} X_0 = \frac{1}{3} \begin{pmatrix} 4 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ -3 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 11 \\ -5 \end{pmatrix}.$$

In coordinates

$$\alpha_1 = \frac{11}{3} \quad \text{and} \quad \alpha_2 = -\frac{5}{3}.$$

The solution to the initial value problem (4.7.11) and (4.7.12) is:

$$\begin{aligned}X(t) &= \frac{1}{3}\left(11e^{2t} v_1 - 5e^{-t} v_2\right) \\ &= \frac{1}{3}\left(11e^{2t} \begin{pmatrix} 1 \\ 1 \end{pmatrix} - 5e^{-t} \begin{pmatrix} 1 \\ 4 \end{pmatrix}\right).\end{aligned}$$

Expressing the solution in coordinates, we obtain:

$$\begin{aligned}x(t) &= \frac{1}{3}\left(11e^{2t} - 5e^{-t}\right) \\ y(t) &= \frac{1}{3}\left(11e^{2t} - 20e^{-t}\right).\end{aligned}$$

An Initial Value Problem Solved using MATLAB Next, solve the system of ODEs

$$\begin{aligned}\dot{x} &= 1.7x + 3.5y \\ \dot{y} &= 1.3x - 4.6y\end{aligned}$$

with initial conditions

$$\begin{aligned}x(0) &= 2.7 \\ y(0) &= 1.1\ .\end{aligned}$$

Rewrite this system in matrix form as

$$\dot{X} = CX$$

where

$$C = \begin{pmatrix} 1.7 & 3.5 \\ 1.3 & -4.6 \end{pmatrix}.$$

Rewrite the initial conditions in vector form

$$X_0 = \begin{pmatrix} 2.7 \\ 1.1 \end{pmatrix}.$$

Now proceed through the four steps outlined previously. In MATLAB begin by typing

```
C  = [1.7 3.5; 1.3 -4.6]
X0 = [2.7; 1.1]
```

Step 1: Find the eigenvalues of C by typing

```
lambda = eig(C)
```

and obtaining

```
lambda =
    2.3543
   -5.2543
```

So the eigenvalues of C are real and distinct.

Step 2: To find the eigenvectors of C we need to solve two homogeneous systems of linear equations. The matrix associated with the first system is obtained by typing

```
C1 = C - lambda(1)*eye(2)
```

which yields

```
C1 =
   -0.6543    3.5000
    1.3000   -6.9543
```

We can solve the homogeneous system (`C1`)$x = 0$ by row reduction — but MATLAB has this process preprogrammed in the command `null`. So type

```
v1 = null(C1)
```

and obtain

```
v1 =
   -0.9830
   -0.1838
```

Similarly, to find an eigenvector associated to the eigenvalue λ_2 type

```
C2 = C - lambda(2)*eye(2);
v2 = null(C2)
```

and obtain

```
v2 =
   -0.4496
    0.8932
```

Step 3: The general solution to this system of differential equations is:

$$X(t) = \alpha_1 e^{2.3543t} \begin{pmatrix} -0.9830 \\ -0.1838 \end{pmatrix} + \alpha_2 e^{-5.2543t} \begin{pmatrix} -0.4496 \\ 0.8932 \end{pmatrix}.$$

Step 4: Solve the initial value problem by finding the scalars α_1 and α_2. Form the matrix A by typing

```
A = [v1 v2]
```

Then solve for the α's by typing

```
alpha = inv(A)*X0
```

obtaining

```
alpha =
   -3.0253
    0.6091
```

Therefore, the closed form solution to the initial value problem is:

$$X(t) = 3.0253e^{2.3543t}\begin{pmatrix} 0.9830 \\ 0.1838 \end{pmatrix}$$
$$+0.6091e^{-5.2543t}\begin{pmatrix} -0.4496 \\ 0.8932 \end{pmatrix}.$$

Exercises

4.8 *Markov Chains

Markov chains provide an interesting and useful application of matrices and linear algebra. In this section we introduce Markov chains via some of the theory and two examples. The theory can be understood and applied to examples using just the background in linear algebra that we have developed in this chapter.

An Example of Cats Consider the four room apartment pictured in Figure 17. One way passages between the rooms are indicated by arrows. For example, it is possible to go from room 1 directly to any other room, but when in room 3 it is possible to go only to room 4.

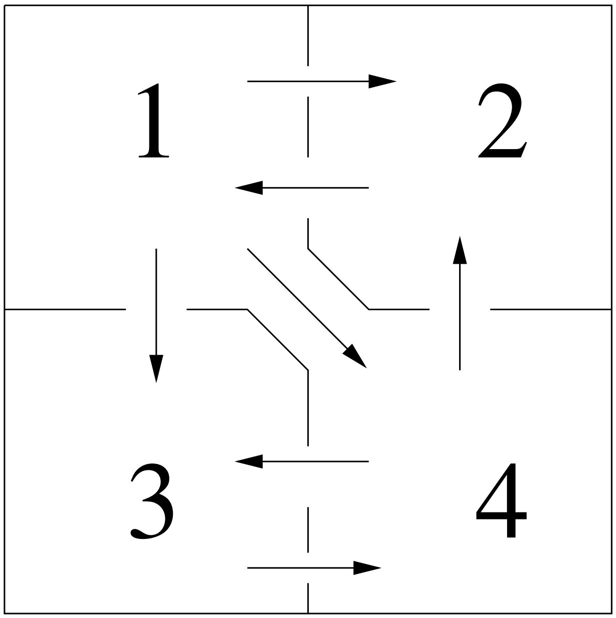

Figure 17: Schematic design of apartment passages.

Suppose that there is a cat in the apartment and that at each hour the cat is asked to move from the room that it is in to another. True to form, however, the cat chooses with equal probability to stay in the room for another hour or to move through one of the allowed passages. Suppose that we let p_{ij} be the probability that the cat will move from room i to room j; in particular, p_{ii} is the probability that the cat will stay in room i. For example, when the cat is in room 1, it has four choices — it can stay in room 1 or move to any of the other rooms. Assuming that each of these choices is made with equal probability, we see that

$$p_{11} = \frac{1}{4} \qquad p_{12} = \frac{1}{4} \qquad p_{13} = \frac{1}{4} \qquad p_{14} = \frac{1}{4}.$$

It is now straightforward to verify that

$$\begin{array}{cccc} p_{21} = \frac{1}{2} & p_{22} = \frac{1}{2} & p_{23} = 0 & p_{24} = 0 \\ p_{31} = 0 & p_{32} = 0 & p_{33} = \frac{1}{2} & p_{34} = \frac{1}{2} \\ p_{41} = 0 & p_{42} = \frac{1}{3} & p_{43} = \frac{1}{3} & p_{44} = \frac{1}{3}. \end{array}$$

Putting these probabilities together yields the *transition matrix*:

$$P = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \qquad (4.8.1^*)$$

This transition matrix has the properties that all entries are nonnegative and that the entries in each row sum to 1.

Three Basic Questions Using the transition matrix P, we discuss the answers to three questions:

(A) What is the probability that a cat starting in room i will be in room j after exactly k steps? We call the movement that occurs after each hour a *step*.

(B) Suppose that we put 100 cats in the apartment with some initial distribution of cats in each room. What will the distribution of cats look like after a large number of steps?

(C) Suppose that a cat is initially in room i and takes a large number of steps. For how many of those steps will the cat be expected to be in room j?

A Discussion of Question (A) We begin to answer Question (A) by determining the probability that the cat moves from room 1 to room 4 in two steps. We denote this probability by $p_{14}^{(2)}$ and compute

$$p_{14}^{(2)} = p_{11}p_{14} + p_{12}p_{24} + p_{13}p_{34} + p_{14}p_{44}; \tag{4.8.2}$$

that is, the probability is the sum of the probabilities that the cat will move from room 1 to each room i and then from room i to room 4. In this case the answer is:

$$p_{14}^{(2)} = \frac{1}{4}\times\frac{1}{4} + \frac{1}{4}\times 0 + \frac{1}{4}\times\frac{1}{2} + \frac{1}{4}\times\frac{1}{3} = \frac{13}{48} \approx 0.27\ .$$

It follows from (4.8.2) and the definition of matrix multiplication that $p_{14}^{(2)}$ is just the $(1,4)^{th}$ entry in the matrix P^2. An induction argument shows that the probability of the cat moving from room i to room j in k steps is precisely the $(i,j)^{th}$ entry in the matrix P^k — which answers Question (A). In particular, we can answer the question: What is the probability that the cat will move from room 4 to room 3 in four steps? Using MATLAB the answer is given by typing `e4_10_1` to recall the matrix P and then typing

```
P4 = P^4;
P4(4,3)
```

obtaining

```
ans =
    0.2728
```

A Discussion of Question (B) We answer Question (B) in two parts: first we compute a formula for determining the number of cats that are expected to be in room i after k steps, and second we explore that formula numerically for large k. We begin by supposing that 100 cats are distributed in the rooms according to the initial vector $V_0 = (v_1, v_2, v_3, v_4)^t$; that is, the number of cats initially in room i is v_i. Next, we denote the number of cats that are expected to be in room i after k steps by $v_i^{(k)}$. For example, we determine how many cats we expect to be in room 2 after one step. That number is:

$$v_2^{(1)} = p_{12}v_1 + p_{22}v_2 + p_{32}v_3 + p_{42}v_4; \tag{4.8.3}$$

that is, $v_2^{(1)}$ is the sum of the proportion of cats in each room i that are expected to migrate to room 2 in one step. In this case, the answer is:

$$\frac{1}{4}v_1 + \frac{1}{2}v_2 + \frac{1}{3}v_4.$$

It now follows from (4.8.3), the definition of the transpose of a matrix, and the definition of matrix multiplication that $v_2^{(1)}$ is the 2^{nd} entry in the vector P^tV_0. Indeed, it follows by induction that $v_i^{(k)}$ is the i^{th} entry in the vector $(P^t)^kV_0$ which answers the first part of Question (B).

We may rephrase the second part of Question (B) as follows. Let

$$V_k = (v_1^k, v_2^k, v_3^k, v_4^k)^t = (P^t)^kV_0.$$

Question (B) actually asks: What will the vector V_k look like for large k. To answer that question we need some results about matrices like the matrix P in (4.8.1*). But first we explore the answer to this question numerically using MATLAB.

Suppose, for example, that the initial vector is

$$V_0 = \begin{pmatrix} 2 \\ 43 \\ 21 \\ 34 \end{pmatrix}. \qquad (4.8.4^*)$$

Typing `e4_10_1` and `e4_10_4` enters the matrix P and the initial vector V_0 into MATLAB. To compute V_{20}, the distribution of cats after 20 steps, type

```
Q=P'
V20 = Q^(20)*V0
```

and obtain

```
V20 =
   18.1818
   27.2727
   27.2727
   27.2727
```

Thus, after rounding to the nearest integer, we expect 27 cats to be in each of rooms 2,3 and 4 and 18 cats to be in room 1 after 20 steps. In fact, the vector V_{20} has a remarkable feature. Compute `Q*V20` in MATLAB and see that $V_{20} = P^t V_{20}$; that is, V_{20} is, to within four digit numerical precision, an eigenvector of P^t with eigenvalue equal to 1. This computation was not a numerical accident, as we now describe. Indeed, compute V_{20} for several initial distributions V_0 of cats and see that the answer will always be the same — up to four digit accuracy.

A Discussion of Question (C) Suppose there is just one cat in the apartment; and we ask how many times that cat is expected to visit room 3 in 100 steps. Suppose the cat starts in room 1; then the initial distribution of cats is one cat in room 1 and zero cats in any of the other rooms. So $V_0 = e_1$. In our discussion of Question (B) we saw that the 3^{rd} entry in $(P^t)^k V_0$ gives the probability c_k that the cat will be in room 3 after k steps.

In the extreme, suppose that the probability that the cat will be in room 3 is 1 for each step k. Then the fraction of the time that the cat is in room 3 is

$$(1 + 1 + \cdots + 1)/100 = 1.$$

In general, the fraction of the time f that the cat will be in room 3 during a span of 100 steps is

$$f = \frac{1}{100}(c_1 + c_2 + \cdots + c_{100}).$$

Since $c_k = (P^t)^k V_0$, we see that

$$f = \frac{1}{100}(P^t V_0 + (P^t)^2 V_0 + \cdots + (P^t)^{100} V_0). \qquad (4.8.5)$$

So, to answer Question (C), we need a way to sum the expression for f in (4.8.5), at least approximately. This is not an easy task — though the answer itself is easy to explain. Let V be the eigenvector of P^t with eigenvalue 1 such that the sum of the entries in V is 1. The answer is: f is approximately equal to V. See Theorem 4.8.4 for a more precise statement.

In our previous calculations the vector V_{20} was seen to be (approximately) an eigenvector of P^t with eigenvalue 1. Moreover the sum of the entries in V_{20} is precisely 100. Therefore, we normalize V_{20} to get V by setting

$$V = \frac{1}{100} V_{20}.$$

So, the fraction of time that the cat spends in room 3 is $f \approx 0.2727$. Indeed, we expect the cat to spend approximately 27% of its time in rooms 2,3,4 and about 18% of its time in room 1.

Markov Matrices We now abstract the salient properties of our cat example. A *Markov chain* is a system with a finite number of states labeled 1,…,n along with probabilities p_{ij} of moving from site i to site j in a single step. The Markov assumption is that these probabilities depend only on the site that you are in and not on how you got there. In our example, we assumed that the probability of the cat moving from say room 2 to room 4 did not depend on how the cat got to room 2 in the first place.

We make a second assumption: there is a k such that it is possible to move from any site i to any site j in exactly k steps. This assumption is *not* valid for general Markov chains, though it is valid for the cat example, since it is possible to move from any room to any other room in that example in exactly three steps. (It takes a minimum of three steps to get from room 3 to room 1 in the cat example.) To simplify our discussion we include this assumption in our definition of a Markov chain.

Definition 4.8.1. *Markov matrices* are square matrices P such that

(a) all entries in P are nonnegative,

(b) the entries in each row of P sum to 1, and

(c) there is a positive integer k such that all of the entries in P^k are positive.

It is straightforward to verify that parts (a) and (b) in the definition of Markov matrices are satisfied by the transition matrix

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \vdots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix}$$

of a Markov chain. To verify part (c) requires further discussion.

Proposition 4.8.2. *Let P be a transition matrix for a Markov chain.*

(a) The probability of moving from site i to site j in exactly k steps is the $(i,j)^{th}$ entry in the matrix P^k.

(b) The expected number of individuals at site i after exactly k steps is the i^{th} entry in the vector $V_k \equiv (P^t)^k V_0$.

(c) P is a Markov matrix.

Proof Only minor changes in our discussion of the cat example proves parts (a) and (b) of the proposition.

(c) The assumption that it is possible to move from each site i to each site j in exactly k steps means that the $(i,j)^{th}$ entry of P^k is positive. For that k, all of the entries of P^k are positive. In the cat example, all entries of P^3 are positive. ■

Proposition 4.8.2 gives the answer to Question (A) and the first part of Question (B) for general Markov chains.

Let $v_i^{(0)} \geq 0$ be the number of individuals initially at site i, and let $V_0 = (v_1^{(0)}, \ldots, v_n^{(0)})^t$. The total number of individuals in the initial population is:

$$\#(V_0) = v_1^{(0)} + \cdots + v_n^{(0)}.$$

Theorem 4.8.3. *Let P be a Markov matrix. Then*

(a) $\#(V_k) = \#(V_0)$; that is, the number of individuals after k time steps is the same as the initial number.

(b) $V = \lim_{k\to\infty} V_k$ exists and $\#(V) = \#(V_0)$.

(c) V is an eigenvector of P^t with eigenvalue equal to 1.

Proof (a) By induction it is sufficient to show that $\#(V_1) = \#(V_0)$. We do this by calculating from $V_1 =$

P^tV_0 that

$$\begin{aligned}\#(V_1) &= v_1^{(1)} + \cdots + v_n^{(1)}\\ &= (p_{11}v_1^{(0)} + \cdots + p_{n1}v_n^{(0)}) + \cdots + (p_{1n}v_1^{(0)} + \cdots + p_{nn}v_n^{(0)})\\ &= (p_{11} + \cdots + p_{1n})v_1^{(0)} + \cdots + (p_{n1} + \cdots + p_{nn})v_n^{(0)}\\ &= v_1^{(0)} + \cdots + v_n^{(0)}\end{aligned}$$

since the entries in each row of P sum to 1. Thus $\#(V_1) = \#(V_0)$, as claimed.

(b) The hard part of this theorem is proving that the limiting vector V exists; we give a proof of this fact in Chapter 11, Theorem 11.4.4. Once V exists it follows directly from (a) that $\#(V) = \#(V_0)$.

(c) Just calculate that

$$\begin{aligned}P^tV &= P^t(\lim_{k\to\infty} V_k) = P^t(\lim_{k\to\infty} (P^t)^kV_0)\\ &= \lim_{k\to\infty} (P^t)^{k+1}V_0 = \lim_{k\to\infty} (P^t)^kV_0 = V,\end{aligned}$$

which proves (c). ■

Theorem 4.8.3(b) gives the answer to the second part of Question (B) for general Markov chains. Next we discuss Question (C).

Theorem 4.8.4. *Let P be a Markov matrix. Let V be the eigenvector of P^t with eigenvalue 1 and $\#(V) = 1$. Then after a large number of steps N the expected number of times an individual will visit site i is Nv_i where v_i is the i^{th} entry in V.*

Sketch of Proof In our discussion of Question (C) for the cat example, we explained why the fraction f_N of time that an individual will visit site j when starting initially at site i is the j^{th} entry in the sum

$$f_N = \frac{1}{N}(P^t + (P^t)^2 + \cdots + (P^t)^N)e_i.$$

See (4.8.5). The proof of this theorem involves being able to calculate the limit of f_N as $N \to \infty$. There are two main ideas. First, the limit of the matrix $(P^t)^N$ exists as N approaches infinity — call that limit Q. Moreover, Q is a matrix all of whose columns equal V. Second, for large N, the sum

$$P^t + (P^t)^2 + \cdots + (P^t)^N \approx Q + Q + \cdots + Q = NQ,$$

so that the limit of the f_N is $Qe_i = V$.

The verification of these statements is beyond the scope of this text. For those interested, the idea of the proof of the second part is roughly the following. Fix k large enough so that $(P^t)^k$ is close to Q. Then when N is large, much larger than k, the sum of the first k terms in the series is nearly zero. ■

Theorem 4.8.4 gives the answer to Question (C) for a general Markov chain. It follows from Theorem 4.8.4 that for Markov chains the amount of time that an individual spends in room i is independent of the individual's initial room — at least after a large number of steps.

A complete proof of this theorem relies on a result known as the *ergodic theorem*. Roughly speaking, the ergodic theorem relates space averages with time averages. To see how this point is relevant, note that Question (B) deals with the issue of how a large number of individuals will be distributed in space after a large number of steps, while Question (C) deals with the issue of how the path of a single individual will be distributed in time after a large number of steps.

An Example of Umbrellas This example focuses on the utility of answering Question (C) and reinforces the fact that results in Theorem 4.8.3 have the second interpretation given in Theorem 4.8.4.

Consider the problem of a man with four umbrellas. If it is raining in the morning when the man is about to leave for his office, then the man takes an umbrella from home to office, assuming that he has an umbrella at home. If it is raining in the afternoon, then the man takes an umbrella from office to home, assuming that he has an umbrella in his office. Suppose that the probability that it will rain in the morning is $p = 0.2$ and the probability that it will rain in the afternoon is $q = 0.3$, and these probabilities are independent. What percentage of days will the man get wet going from home to office; that is, what percentage of the days will the man be at home on a rainy morning with all of his umbrellas at the office?

There are five states in the system depending on the number of umbrellas that are at home. Let s_i where $0 \leq i \leq 4$ be the state with i umbrellas at home and $4-i$ umbrellas at work. For example, s_2 is the state of having two umbrellas at home and two at the office. Let P be the 5×5 transition matrix of state changes from morning to afternoon and Q be the 5×5 transition matrix of state changes from afternoon to morning. For example, the probability p_{23} of moving from site s_2 to site s_3 is 0, since it is not possible to have more umbrellas at home after going to work in the morning. The probability $q_{23} = q$, since the number of umbrellas at home will increase by one only if it is raining in the afternoon. The transition probabilities between all states are given in the following transition matrices:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ p & 1-p & 0 & 0 & 0 \\ 0 & p & 1-p & 0 & 0 \\ 0 & 0 & p & 1-p & 0 \\ 0 & 0 & 0 & p & 1-p \end{pmatrix};$$

$$Q = \begin{pmatrix} 1-q & q & 0 & 0 & 0 \\ 0 & 1-q & q & 0 & 0 \\ 0 & 0 & 1-q & q & 0 \\ 0 & 0 & 0 & 1-q & q \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Specifically,

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.2 & 0.8 & 0 & 0 & 0 \\ 0 & 0.2 & 0.8 & 0 & 0 \\ 0 & 0 & 0.2 & 0.8 & 0 \\ 0 & 0 & 0 & 0.2 & 0.8 \end{pmatrix} \qquad (4.8.6^*)$$

$$Q = \begin{pmatrix} 0.7 & 0.3 & 0 & 0 & 0 \\ 0 & 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0 & 0.7 & 0.3 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The transition matrix M from moving from state s_i on one morning to state s_j the next morning is just $M = PQ$. We can compute this matrix using MATLAB by typing

```
e4_10_6
M = P*Q
```

obtaining

```
M =
    0.7000    0.3000         0         0         0
    0.1400    0.6200    0.2400         0         0
         0    0.1400    0.6200    0.2400         0
         0         0    0.1400    0.6200    0.2400
         0         0         0    0.1400    0.8600
```

It is easy to check using MATLAB that all entries in the matrix M^4 are nonzero. So M is a Markov matrix and we can use Theorem 4.8.4 to find the limiting distribution of states. Start with some initial condition like $V_0 = (0,0,1,0,0)^t$ corresponding to the state in which two umbrellas are at home and two at the office. Then compute the vectors $V_k = (M^t)^k V_0$ until arriving at an eigenvector of M^t with eigenvalue 1. For example, V_{70} is computed by typing `V70 = M'^(70)*V0` and obtaining

```
V70 =
    0.0419
    0.0898
    0.1537
    0.2633
    0.4512
```

We interpret $V \approx V_{70}$ in the following way. Since v_1 is approximately .042, it follows that for approximately 4.2% of all steps the umbrellas are in state s_0. That is, approximately 4.2% of all days there are no umbrellas at home. The probability that it will rain in the morning on one of those days is 0.2. Therefore, the probability of being at home in the morning when it is raining without any umbrellas is approximately 0.008.

Exercises

5 Vector Spaces

In Chapter 2 we discussed how to solve systems of m linear equations in n unknowns. We found that solutions of these equations are vectors $(x_1, \ldots, x_n) \in \mathbb{R}^n$. In Chapter 3 we discussed how the notation of matrices and matrix multiplication drastically simplifies the presentation of linear systems and how matrix multiplication leads to linear mappings. We also discussed briefly how linear mappings lead to methods for solving linear systems — superposition, eigenvectors, inverses. In Chapter 4 we discussed how to solve systems of n linear differential equations in n unknown functions. These chapters have provided an introduction to many of the ideas of linear algebra and now we begin the task of formalizing these ideas.

Sets having the two operations of vector addition and scalar multiplication are called *vector spaces*. This concept is introduced in Section 5.1 along with the two primary examples — the set $\mathbb{R}^n$ in which solutions to systems of linear equations sit and the set $\mathcal{C}^1$ of differentiable functions in which solutions to systems of ordinary differential equations sit. Solutions to systems of homogeneous linear equations form subspaces of $\mathbb{R}^n$ and solutions of systems of linear differential equations form subspaces of $\mathcal{C}^1$. These issues are discussed in Sections 5.1 and 5.2.

When we *solve* a homogeneous system of equations, we write every solution as a superposition of a finite number of specific solutions. Abstracting this process is one of the main points of this chapter. Specifically, we show that every vector in many commonly occurring vector spaces (in particular, the subspaces of solutions) can be written as a *linear combination* (superposition) of a few solutions. The minimum number of solutions needed is called the *dimension* of that vector space. Sets of vectors that generate all solutions by superposition and that consist of that minimum number of vectors are called *bases*. These ideas are discussed in detail in Sections 5.3–5.5. The proof of the main theorem (Theorem 5.5.3), which gives a computable method for determining when a set is a basis, is given in Section 5.6. This proof may be omitted on a first reading, but the statement of the theorem is most important and must be understood.

5.1 Vector Spaces and Subspaces

Vector spaces abstract the arithmetic properties of addition and scalar multiplication of vectors. In $\mathbb{R}^n$ we know how to add vectors and to multiply vectors by scalars. Indeed, it is straightforward to verify that each of the eight properties listed in Table 1 is valid for vectors in $V = \mathbb{R}^n$. Remarkably, sets that satisfy these eight properties have much in common with $\mathbb{R}^n$. So we define:

Definition 5.1.1. Let V be a set having the two operations of addition and scalar multiplication. Then V is a *vector space* if the eight properties listed in Table 5.1.1 hold. The elements of a vector space are called *vectors*.

The vector 0 mentioned in (A3) in Table 1 is called the *zero vector*.

When we say that a vector space V has the two operations of addition and scalar multiplication we mean that the sum of two vectors in V is again a vector in V and the scalar product of a vector with a number is again a vector in V. These two properties are called *closure under addition* and *closure under scalar multiplication*.

In this discussion we focus on just two types of vector spaces: $\mathbb{R}^n$ and function spaces. The reason that we make this choice is that solutions to linear equations are vectors in $\mathbb{R}^n$ while solutions to linear systems of differential equations are vectors of functions.

An Example of a Function Space

For example, let $\mathcal{F}$ denote the set of all functions $f : \mathbb{R} \to \mathbb{R}$. Note that functions like $f_1(t) = t^2 - 2t + 7$ and $f_2(t) = \sin t$ are in $\mathcal{F}$ since they are defined for all real numbers t, but that functions like $g_1(t) = \dfrac{1}{t}$ and $g_2(t) = \tan t$ are not in $\mathcal{F}$ since they are not defined for all t.

We can add two functions f and g by defining the function $f + g$ to be:

$$(f + g)(t) = f(t) + g(t).$$

We can also multiply a function f by a scalar $c \in \mathbb{R}$ by defining the function cf to be:

$$(cf)(t) = cf(t).$$

With these operations of addition and scalar multiplication, $\mathcal{F}$ is a vector space; that is, $\mathcal{F}$ satisfies the eight vector space properties in Table 1. More precisely:

(A3) Define the zero function $\mathcal{O}$ by

$$\mathcal{O}(t) = 0 \quad \text{for all } t \in \mathbb{R}.$$

For every x in $\mathcal{F}$ the function $\mathcal{O}$ satisfies:

$$(x + \mathcal{O})(t) = x(t) + \mathcal{O}(t) = x(t) + 0 = x(t).$$

Therefore, $x + \mathcal{O} = x$ and $\mathcal{O}$ is the additive identity in $\mathcal{F}$.

(A4) Let x be a function in $\mathcal{F}$ and define $y(t) = -x(t)$. Then y is also a function in $\mathcal{F}$, and

$$(x+y)(t) = x(t)+y(t) = x(t)+(-x(t)) = 0 = \mathcal{O}(t).$$

Thus, x has the additive inverse $-x$.

After these comments it is straightforward to verify that the remaining six properties in Table 1 are satisfied by functions in $\mathcal{F}$.

Sets that are not Vector Spaces

It is worth considering how closure under vector addition and scalar multiplication can fail. Consider the following three examples.

(i) Let V_1 be the set that consists of just the x and y axes in the plane. Since $(1, 0)$ and $(0, 1)$ are in V_1 but

$$(1, 0) + (0, 1) = (1, 1)$$

is not in V_1, we see that V_1 is not closed under vector addition. On the other hand, V_1 is closed under scalar multiplication.

Table 1: Properties of Vector Spaces: suppose $u, v, w \in V$ and $r, s \in \mathbb{R}$.

(A1)	Addition is commutative	$v + w = w + v$
(A2)	Addition is associative	$(u + v) + w = u + (v + w)$
(A3)	Additive identity 0 exists	$v + 0 = v$
(A4)	Additive inverse $-v$ exists	$v + (-v) = 0$
(M1)	Multiplication is associative	$(rs)v = r(sv)$
(M2)	Multiplicative identity exists	$1v = v$
(D1)	Distributive law for scalars	$(r + s)v = rv + sv$
(D2)	Distributive law for vectors	$r(v + w) = rv + rw$

(ii) Let V_2 be the set of all vectors $(k, \ell) \in \mathbb{R}^2$ where k and ℓ are integers. The set V_2 is closed under addition but not under scalar multiplication since $\frac{1}{2}(1, 0) = (\frac{1}{2}, 0)$ is not in V_2.

(iii) Let $V_3 = [1, 2]$ be the closed interval in $\mathbb{R}$. The set V_3 is neither closed under addition ($1 + 1.5 = 2.5 \notin V_3$) nor under scalar multiplication ($4 \cdot 1.5 = 6 \notin V_3$). Hence the set V_3 is not closed under vector addition and not closed under scalar multiplication.

Subspaces

Definition 5.1.2. Let V be a vector space. A nonempty subset $W \subset V$ is a *subspace* if W is a vector space using the operations of addition and scalar multiplication defined on V.

Note that in order for a subset W of a vector space V to be a subspace it must be *closed under addition* and *closed under scalar multiplication*. That is, suppose $w_1, w_2 \in W$ and $r \in \mathbb{R}$. Then

(i) $w_1 + w_2 \in W$, and

(ii) $rw_1 \in W$.

The x-axis and the xz-plane are examples of subsets of $\mathbb{R}^3$ that are closed under addition and closed under scalar multiplication. Every vector on the x-axis has the form $(a, 0, 0) \in \mathbb{R}^3$. The sum of two vectors $(a, 0, 0)$ and $(b, 0, 0)$ on the x-axis is $(a + b, 0, 0)$ which is also on the x-axis. The x-axis is also closed under scalar multiplication as $r(a, 0, 0) = (ra, 0, 0)$, and the x-axis is a subspace of $\mathbb{R}^3$. Similarly, every vector in the xz-plane in $\mathbb{R}^3$ has the form $(a_1, 0, a_3)$. As in the case of the x-axis, it is easy to verify that this set of vectors is closed under addition and scalar multiplication. Thus, the xz-plane is also a subspace of $\mathbb{R}^3$.

In Theorem 5.1.4 we show that every subset of a vector space that is closed under addition and scalar multiplication is a subspace. To verify this statement, we need the following lemma in which some special notation is used. Typically, we use the same notation 0 to denote the real number zero and the zero vector. In the following lemma it is convenient to distinguish the two different uses of 0, and we write the zero vector in boldface.

Lemma 5.1.3. *Let V be a vector space, and let $\mathbf{0} \in V$ be the zero vector. Then*

$$0v = \mathbf{0} \quad \textit{and} \quad (-1)v = -v$$

for every vector in $v \in V$.

Proof Let v be a vector in V and use (D1) to compute

$$0v + 0v = (0 + 0)v = 0v.$$

By (A4) the vector $0v$ has an additive inverse $-0v$. Adding $-0v$ to both sides yields

$$(0v + 0v) + (-0v) = 0v + (-0v) = \mathbf{0}.$$

Associativity of addition (A2) now implies

$$0v + (0v + (-0v)) = \mathbf{0}.$$

A second application of (A4) implies that

$$0v + \mathbf{0} = \mathbf{0}$$

and (A3) implies that $0v = \mathbf{0}$.

Next, we show that the additive inverse $-v$ of a vector v is unique. That is, if $v + a = \mathbf{0}$, then $a = -v$.

Before beginning the proof, note that commutativity of addition (A1) together with (A3) implies that $\mathbf{0} + v = v$. Similarly, (A1) and (A4) imply that $-v + v = \mathbf{0}$.

To prove uniqueness of additive inverses, add $-v$ to both sides of the equation $v + a = \mathbf{0}$ yielding

$$-v + (v + a) = -v + \mathbf{0}.$$

Properties (A2) and (A3) imply

$$(-v + v) + a = -v.$$

But

$$(-v + v) + a = \mathbf{0} + a = a.$$

Therefore $a = -v$, as claimed.

To verify that $(-1)v = -v$, we show that $(-1)v$ is the additive inverse of v. Using (M1), (D1), and the fact that $0v = \mathbf{0}$, calculate

$$v + (-1)v = 1v + (-1)v = (1 - 1)v = 0v = \mathbf{0}.$$

Thus, $(-1)v$ is the additive inverse of v and must equal $-v$, as claimed. ■

Theorem 5.1.4. *Let W be a subset of the vector space V. If W is closed under addition and closed under scalar multiplication, then W is a subspace.*

Proof We have to show that W is a vector space using the operations of addition and scalar multiplication defined on V. That is, we need to verify that the eight properties listed in Table 1 are satisfied. Note that properties (A1), (A2), (M1), (M2), (D1), and (D2) are valid for vectors in W since they are valid for vectors in V.

It remains to verify (A3) and (A4). Let $w \in W$ be any vector. Since W is closed under scalar multiplication, it follows that $0w$ and $(-1)w$ are in W. Lemma 5.1.3 states that $0w = 0$ and $(-1)w = -w$; it follows that 0 and $-w$ are in W. Hence, properties (A3) and (A4) are valid for vectors in W, since they are valid for vectors in V. ■

Examples of Subspaces of $\mathbb{R}^n$

Example 5.1.5. (a) Let V be a vector space. Then the subsets V and $\{0\}$ are always subspaces of V. A subspace $W \subset V$ is *proper* if $W \neq 0$ and $W \neq V$.

(b) Lines through the origin are subspaces of $\mathbb{R}^n$. Let $w \in \mathbb{R}^n$ be a nonzero vector and let $W = \{rw : r \in \mathbb{R}\}$. The set W is closed under addition and scalar multiplication and is a subspace of $\mathbb{R}^n$ by Theorem 5.1.4. The subspace W is just a *line through the origin* in $\mathbb{R}^n$, since the vector rw points in the same direction as w when $r > 0$ and the exact opposite direction when $r < 0$.

(c) Planes containing the origin are subspaces of $\mathbb{R}^3$. To verify this point, let P be a plane through the origin and let N be a vector perpendicular to P. Then P consists of all vectors $v \in \mathbb{R}^3$ perpendicular to N; using the dot-product (see Chapter 2, (2.2.3)) we recall that such vectors satisfy the linear equation $N \cdot v = 0$. By superposition, the set of all solutions to this equation is closed under addition and scalar multiplication and is therefore a subspace by Theorem 5.1.4.

In a sense that will be made precise all subspaces of $\mathbb{R}^n$ can be written as the span of a finite number of vectors generalizing Example 5.1.5(b) or as solutions to a system of linear equations generalizing Example 5.1.5(c).

Examples of Subspaces of the Function Space $\mathcal{F}$ Let $\mathcal{P}$ be the set of all polynomials in $\mathcal{F}$. The sum of two polynomials is a polynomial and the scalar multiple of a polynomial is a polynomial. Thus, $\mathcal{P}$ is closed under addition and scalar multiplication, and $\mathcal{P}$ is a subspace of $\mathcal{F}$.

As a second example of a subspace of $\mathcal{F}$, let $\mathcal{C}^1$ be the set of all continuously differentiable functions $u : \mathbb{R} \to \mathbb{R}$. A function u is in $\mathcal{C}^1$ if u and u' exist and are continuous for all $t \in \mathbb{R}$. Examples of functions in $\mathcal{C}^1$ are:

(i) Every polynomial $p(t) = a_m t^m + a_{m-1}t^{m-1} + \cdots + a_1 t + a_0$ is in $\mathcal{C}^1$.

(ii) The function $u(t) = e^{\lambda t}$ is in $\mathcal{C}^1$ for each constant $\lambda \in \mathbb{R}$.

(iii) The trigonometric functions $u(t) = \sin(\lambda t)$ and $v(t) = \cos(\lambda t)$ are in $\mathcal{C}^1$ for each constant $\lambda \in \mathbb{R}$.

(iv) $u(t) = t^{7/3}$ is twice differentiable everywhere and is in $\mathcal{C}^1$.

Equally there are many commonly used functions that are not in $\mathcal{C}^1$. Examples include:

(i) $u(t) = \dfrac{1}{t-5}$ is neither defined nor continuous at $t = 5$.

(ii) $u(t) = |t|$ is not differentiable (at $t = 0$).

(iii) $u(t) = \csc(t)$ is neither defined nor continuous at $t = k\pi$ for any integer k.

The subset $\mathcal{C}^1 \subset \mathcal{F}$ is a subspace and hence a vector space. The reason is simple. If $x(t)$ and $y(t)$ are continuously differentiable, then

$$\frac{d}{dt}(x+y) = \frac{dx}{dt} + \frac{dy}{dt}.$$

Hence $x + y$ is differentiable and is in $\mathcal{C}^1$ and $\mathcal{C}^1$ is closed under addition. Similarly, $\mathcal{C}^1$ is closed under scalar multiplication. Let $r \in \mathbb{R}$ and let $x \in \mathcal{C}^1$. Then

$$\frac{d}{dt}(rx)(t) = r\frac{dx}{dt}(t).$$

Hence rx is differentiable and is in $\mathcal{C}^1$.

The Vector Space $(\mathcal{C}^1)^n$ Another example of a vector space that combines the features of both $\mathbb{R}^n$ and $\mathcal{C}^1$ is $(\mathcal{C}^1)^n$. Vectors $u \in (\mathcal{C}^1)^n$ have the form

$$u(t) = (u_1(t), \ldots, u_n(t)),$$

where each coordinate function $u_j(t) \in \mathcal{C}^1$. Addition and scalar multiplication in $(\mathcal{C}^1)^n$ are defined coordinatewise — just like addition and scalar multiplication in $\mathbb{R}^n$. That is, let u, v be in $(\mathcal{C}^1)^n$ and let r be in $\mathbb{R}$, then

$$\begin{array}{rcl} (u+v)(t) & = & (u_1(t)+v_1(t), \ldots, u_n(t)+v_n(t)) \\ (ru)(t) & = & (ru_1(t), \ldots, ru_n(t)). \end{array}$$

The set $(\mathcal{C}^1)^n$ satisfies the eight properties of vector spaces and is a vector space. Solutions to systems of n linear ordinary differential equations are vectors in $(\mathcal{C}^1)^n$.

Exercises

5.2 Construction of Subspaces

The principle of superposition shows that the set of all solutions to a homogeneous system of linear equations is closed under addition and scalar multiplication and is a subspace. Indeed, there are two ways to describe subspaces: first as solutions to linear systems, and second as the span of a set of vectors. We shall see that solving a homogeneous linear system of equations just means writing the solution set as the span of a finite set of vectors.

Solutions to Homogeneous Systems Form Subspaces

Definition 5.2.1. Let A be an $m \times n$ matrix. The *null space* of A is the set of solutions to the homogeneous system of linear equations

$$Ax = 0. \tag{5.2.1}$$

Lemma 5.2.2. *Let A be an $m \times n$ matrix. Then the null space of A is a subspace of $\mathbb{R}^n$.*

Proof Suppose that x and y are solutions to (5.2.1). Then

$$A(x+y) = Ax + Ay = 0 + 0 = 0;$$

so $x + y$ is a solution of (5.2.1). Similarly, for $r \in \mathbb{R}$

$$A(rx) = rAx = r0 = 0;$$

so rx is a solution of (5.2.1). Thus, $x + y$ and rx are in the null space of A, and the null space is closed under addition and scalar multiplication. So Theorem 5.1.4 implies that the null space is a subspace of the vector space $\mathbb{R}^n$. ■

Solutions to Linear Systems of Differential Equations Form Subspaces Let C be an $n \times n$ matrix and let W be the set of solutions to the linear system of ordinary differential equations

$$\frac{dx}{dt}(t) = Cx(t). \tag{5.2.2}$$

We will see later that a solution to (5.2.2) has coordinate functions $x_j(t)$ in $\mathcal{C}^1$. The principle of superposition then shows that W is a subspace of $(\mathcal{C}^1)^n$. Suppose $x(t)$ and $y(t)$ are solutions of (5.2.2). Then

$$\frac{d}{dt}(x(t)+y(t)) = \frac{dx}{dt}(t)+\frac{dy}{dt}(t) = Cx(t)+Cy(t) = C(x(t)+y(t));$$

so $x(t)+y(t)$ is a solution of (5.2.2) and in W. A similar calculation shows that $rx(t)$ is also in W and that $W \subset (\mathcal{C}^1)^n$ is a subspace.

Writing Solution Subspaces as a Span The way we solve homogeneous systems of equations gives a second method for defining subspaces. For example, consider the system

$$Ax = 0,$$

where

$$A = \begin{pmatrix} 2 & 1 & 4 & 0 \\ -1 & 0 & 2 & 1 \end{pmatrix}.$$

The matrix A is row equivalent to the reduced echelon form matrix

$$E = \begin{pmatrix} 1 & 0 & -2 & -1 \\ 0 & 1 & 8 & 2 \end{pmatrix}.$$

Therefore $x = (x_1, x_2, x_3, x_4)$ is a solution of $Ex = 0$ if and only if $x_1 = 2x_3 + x_4$ and $x_2 = -8x_3 - 2x_4$. It follows that every solution of $Ex = 0$ can be written as:

$$x = x_3 \begin{pmatrix} 2 \\ -8 \\ 1 \\ 0 \end{pmatrix} + x_4 \begin{pmatrix} 1 \\ -2 \\ 0 \\ 1 \end{pmatrix}.$$

Since row operations do not change the set of solutions, it follows that every solution of $Ax = 0$ has this form. We have also shown that every solution is generated by two vectors by use of vector addition and scalar multiplication. We say that this subspace is *spanned* by the two vectors

$$\begin{pmatrix} 2 \\ -8 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 \\ -2 \\ 0 \\ 1 \end{pmatrix}.$$

For example, a calculation verifies that the vector

$$\begin{pmatrix} -1 \\ -2 \\ 1 \\ -3 \end{pmatrix}$$

is also a solution of $Ax = 0$. Indeed, we may write it as

$$\begin{pmatrix} -1 \\ -2 \\ 1 \\ -3 \end{pmatrix} = \begin{pmatrix} 2 \\ -8 \\ 1 \\ 0 \end{pmatrix} - 3 \begin{pmatrix} 1 \\ -2 \\ 0 \\ 1 \end{pmatrix}. \tag{5.2.3}$$

Spans Let $v_1, \ldots, v_k$ be a set of vectors in a vector space V. A vector $v \in V$ is a *linear combination* of $v_1, \ldots, v_k$ if

$$v = r_1 v_1 + \cdots + r_k v_k$$

for some scalars $r_1, \ldots, r_k$.

Definition 5.2.3. The set of all linear combinations of the vectors $v_1, \ldots, v_k$ in a vector space V is the *span* of $v_1, \ldots, v_k$ and is denoted by $\mathrm{span}\{v_1, \ldots, v_k\}$.

For example, the vector on the left hand side in (5.2.3) is a linear combination of the two vectors on the right hand side.

The simplest example of a span is $\mathbb{R}^n$ itself. Let $v_j = e_j$ where $e_j \in \mathbb{R}^n$ is the vector with a 1 in the j^{th} coordinate and 0 in all other coordinates. Then every vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ can be written as

$$x = x_1 e_1 + \cdots + x_n e_n.$$

It follows that

$$\mathbb{R}^n = \mathrm{span}\{e_1, \ldots, e_n\}.$$

Similarly, the set $\mathrm{span}\{e_1, e_3\} \subset \mathbb{R}^3$ is just the x_1x_3-plane, since vectors in this span are

$$x_1 e_1 + x_3 e_3 = x_1(1, 0, 0) + x_3(0, 0, 1) = (x_1, 0, x_3).$$

Proposition 5.2.4. *Let V be a vector space and let $w_1, \ldots, w_k \in V$. Then $W = \mathrm{span}\{w_1, \ldots, w_k\} \subset V$ is a subspace.*

Proof Suppose $x, y \in W$. Then

$$\begin{aligned} x &= r_1 w_1 + \cdots + r_k w_k \\ y &= s_1 w_1 + \cdots + s_k w_k \end{aligned}$$

for some scalars $r_1, \ldots, r_k$ and $s_1, \ldots, s_k$. It follows that

$$x + y = (r_1 + s_1) w_1 + \cdots + (r_k + s_k) w_k$$

and

$$rx = (rr_1) w_1 + \cdots + (rr_k) w_k$$

are both in $\mathrm{span}\{w_1, \ldots, w_k\}$. Hence $W \subset V$ is closed under addition and scalar multiplication, and is a subspace by Theorem 5.1.4. ■

For example, let

$$v = (2, 1, 0) \quad \text{and} \quad w = (1, 1, 1) \tag{5.2.4}$$

be vectors in $\mathbb{R}^3$. Then linear combinations of the vectors v and w have the form

$$\alpha v + \beta w = (2\alpha + \beta, \alpha + \beta, \beta)$$

for real numbers α and β. Note that every one of these vectors is a solution to the linear equation

$$x_1 - 2x_2 + x_3 = 0, \tag{5.2.5}$$

that is, the 1^{st} coordinate minus twice the 2^{nd} coordinate plus the 3^{rd} coordinate equals zero. Moreover, you may verify that every solution of (5.2.5) is a linear combination of the vectors v and w in (5.2.4). Thus, the set of solutions to the homogeneous linear equation (5.2.5) is a subspace, and that subspace can be written as the span of all linear combinations of the vectors v and w.

In this language we see that the process of solving a homogeneous system of linear equations is just the process of finding a set of vectors that span the subspace of all solutions. Indeed, we can now restate Theorem 2.4.6 of Chapter 2. Recall that a matrix A has *rank* ℓ if it is row equivalent to a matrix in echelon form with ℓ nonzero rows.

Proposition 5.2.5. *Let A be an $m \times n$ matrix with rank ℓ. Then the null space of A is the span of $n - \ell$ vectors.*

We have now seen that there are two ways to describe subspaces — as solutions of homogeneous systems of linear equations and as a span of a set of vectors, the *spanning set*. Much of linear algebra is concerned with determining how one goes from one description of a subspace to the other.

Exercises

5.3 Spanning Sets and MATLAB

In this section we discuss:

- how to find a spanning set for the subspace of solutions to a homogeneous system of linear equations using the MATLAB command `null`, and
- how to determine when a vector is in the subspace spanned by a set of vectors using the MATLAB command `rref`.

Spanning Sets for Homogeneous Linear Equations In Chapter 2 we saw how to use Gaussian elimination, back substitution, and MATLAB to compute solutions to a system of linear equations. For systems of homogeneous equations, MATLAB provides a command to find a spanning set for the subspace of solutions. That command is `null`. For example, if we type

```
A = [2 1 4 0; -1 0 2 1]
B = null(A)
```

then we obtain

```
B =
    0.4830         0
   -0.4140    0.8729
   -0.1380   -0.2182
    0.7591    0.4364
```

The two columns of the matrix B span the set of solutions of the equation $Ax = 0$. In particular, the vector $(2, -8, 1, 0)$ is a solution to $Ax = 0$ and is therefore a linear combination of the column vectors of `B`. Indeed, type

```
4.1404*B(:,1)-7.2012*B(:,2)
```

and observe that this linear combination is the desired one.

Next we describe how to find the coefficients `4.1404` and `-7.2012` by showing that these coefficients themselves are solutions to another system of linear equations.

When is a Vector in a Span? Let $w_1, \ldots, w_k$ and v be vectors in $\mathbb{R}^n$. We now describe a method that allows us to decide whether v is in $\mathrm{span}\{w_1, \ldots, w_k\}$. To answer this question one has to solve a system of n linear equations in k unknowns. The unknowns correspond to the coefficients in the linear combination of the vectors $w_1, \ldots, w_k$ that gives v.

Let us be more precise. The vector v is in $\mathrm{span}\{w_1, \ldots, w_k\}$ if and only if there are constants $r_1, \ldots, r_k$ such that the equation

$$r_1 w_1 + \cdots + r_k w_k = v \tag{5.3.1}$$

is valid. Define the $n \times k$ matrix A as the one having $w_1, \ldots, w_k$ as its columns; that is,

$$A = (w_1 | \cdots | w_k). \tag{5.3.2}$$

Let r be the k-vector

$$r = \begin{pmatrix} r_1 \\ \vdots \\ r_k \end{pmatrix}.$$

Then we may rewrite equation (5.3.1) as

$$Ar = v. \tag{5.3.3}$$

To summarize:

Lemma 5.3.1. *Let $w_1, \ldots, w_k$ and v be vectors in $\mathbb{R}^n$. Then v is in $\mathrm{span}\{w_1, \ldots, w_k\}$ if and only if the system of linear equations (5.3.3) has a solution where A is the $n \times k$ defined in (5.3.2).*

To solve (5.3.3) we row reduce the augmented matrix $(A|v)$. For example, is $v = (2,1)$ in the span of $w_1 = (1,1)$ and $w_2 = (1,-1)$? That is, do there exist scalars r_1, r_2 such that

$$r_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + r_2 \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}?$$

As noted, we can rewrite this equation as

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

We can solve this equation by row reducing the augmented matrix

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ 1 & -1 & 1 \end{array}\right)$$

to obtain

$$\left(\begin{array}{cc|c} 1 & 0 & \frac{3}{2} \\ 0 & 1 & \frac{1}{2} \end{array}\right).$$

So $v = \frac{3}{2}w_1 + \frac{1}{2}w_2$.

Row reduction to reduced echelon form has been preprogrammed in the MATLAB command `rref`. Consider the following example. Let

$$w_1 = (2,0,-1,4) \quad \text{and} \quad w_2 = (2,-1,0,2) \qquad (5.3.4)$$

and ask the question whether $v = (-2,4,-3,4)$ is in $\mathrm{span}\{w_1, w_2\}$.

In MATLAB load the matrix A having w_1 and w_2 as its columns and the vector v by typing `e5_3_5`

$$A = \begin{pmatrix} 2 & 2 \\ 0 & -1 \\ -1 & 0 \\ 4 & 2 \end{pmatrix} \quad \text{and} \quad v = \begin{pmatrix} -2 \\ 4 \\ -3 \\ 4 \end{pmatrix}. \qquad (5.3.5^*)$$

We can solve the system of equations using MATLAB. First, form the augmented matrix by typing

```
aug = [A v]
```

Then solve the system by typing `rref(aug)` to obtain

```
ans =
     1     0     3
     0     1    -4
     0     0     0
     0     0     0
```

It follows that $(r_1, r_2) = (3,-4)$ is a solution and $v = 3w_1 - 4w_2$.

Now we change the 4^{th} entry in v slightly by typing `v(4) = 4.01`. There is no solution to the system of equations

$$Ar = \begin{pmatrix} -2 \\ 4 \\ -3 \\ 4.01 \end{pmatrix}$$

as we now show. Type

```
aug = [A v]
rref(aug)
```

which yields

```
ans =
     1     0     0
     0     1     0
     0     0     1
     0     0     0
```

This matrix corresponds to an inconsistent system; thus v is no longer in the span of w_1 and w_2.

Exercises

5.4 Linear Dependence and Linear Independence

An important question in linear algebra concerns finding spanning sets for subspaces having the smallest number of vectors. Let $w_1,\ldots,w_k$ be vectors in a vector space V and let $W = \mathrm{span}\{w_1,\ldots,w_k\}$. Suppose that W is generated by a subset of these k vectors. Indeed, suppose that the k^{th} vector is redundant in the sense that $W = \mathrm{span}\{w_1,\ldots,w_{k-1}\}$. Since $w_k \in W$, this is possible only if w_k is a linear combination of the $k-1$ vectors $w_1,\ldots,w_{k-1}$; that is, only if

$$w_k = r_1w_1 + \cdots + r_{k-1}w_{k-1}. \tag{5.4.1}$$

Definition 5.4.1. Let $w_1,\ldots,w_k$ be vectors in the vector space V. The set $\{w_1,\ldots,w_k\}$ is *linearly dependent* if one of the vectors w_j can be written as a linear combination of the remaining $k-1$ vectors.

Note that when $k=1$, the phrase '$\{w_1\}$ is linearly dependent' means that $w_1 = 0$.

If we set $r_k = -1$, then we may rewrite (5.4.1) as

$$r_1w_1 + \cdots + r_{k-1}w_{k-1} + r_kw_k = 0.$$

It follows that:

Lemma 5.4.2. *The set of vectors $\{w_1,\ldots,w_k\}$ is linearly dependent if and only if there exist scalars $r_1,\ldots,r_k$ such that*

(a) at least one of the r_j is nonzero, and

(b) $r_1w_1 + \cdots + r_kw_k = 0$.

For example, the vectors $w_1 = (2,4,7)$, $w_2 = (5,1,-1)$, and $w_3 = (1,-7,-15)$ are linearly dependent since $2w_1 - w_2 + w_3 = 0$.

Definition 5.4.3. A set of k vectors $\{w_1,\ldots,w_k\}$ is *linearly independent* if none of the k vectors can be written as a linear combination of the other $k-1$ vectors.

Since linear independence means *not* linearly dependent, Lemma 5.4.2 can be rewritten as:

Lemma 5.4.4. *The set of vectors $\{w_1,\ldots,w_k\}$ is linearly independent if and only if whenever*

$$r_1w_1 + \cdots + r_kw_k = 0,$$

it follows that

$$r_1 = r_2 = \cdots = r_k = 0.$$

Let e_j be the vector in $\mathbb{R}^n$ whose j^{th} component is 1 and all of whose other components are 0. The set of vectors $e_1,\ldots,e_n$ is the simplest example of a set of linearly independent vectors in $\mathbb{R}^n$. We use Lemma 5.4.4 to verify independence by supposing that

$$r_1e_1 + \cdots + r_ne_n = 0.$$

A calculation shows that

$$0 = r_1e_1 + \cdots + r_ne_n = (r_1,\ldots,r_n).$$

It follows that each r_j equals 0, and the vectors $e_1,\ldots,e_n$ are linearly independent.

Deciding Linear Dependence and Linear Independence

Deciding whether a set of k vectors in $\mathbb{R}^n$ is linearly dependent or linearly independent is equivalent to solving a system of linear equations. Let $w_1,\ldots,w_k$ be vectors in $\mathbb{R}^n$, and view these vectors as column vectors. Let

$$A = (w_1|\cdots|w_k) \tag{5.4.2}$$

be the $n \times k$ matrix whose columns are the vectors w_j. Then a vector

$$R = \begin{pmatrix} r_1 \\ \vdots \\ r_k \end{pmatrix}$$

is a solution to the system of equations $AR = 0$ precisely when

$$r_1w_1 + \cdots + r_kw_k = 0. \tag{5.4.3}$$

If there is a nonzero solution R to $AR = 0$, then the vectors $\{w_1, \ldots, w_k\}$ are linearly dependent; if the only solution to $AR = 0$ is $R = 0$, then the vectors are linearly independent.

The preceding discussion is summarized by:

Lemma 5.4.5. *The vectors $w_1, \ldots, w_k$ in $\mathbb{R}^n$ are linearly dependent if the null space of the $n \times k$ matrix A defined in (5.4.2) is nonzero and linearly independent if the null space of A is zero.*

A Simple Example of Linear Independence with Two Vectors

The two vectors

$$w_1 = \begin{pmatrix} 2 \\ -8 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad w_2 = \begin{pmatrix} 1 \\ -2 \\ 0 \\ 1 \end{pmatrix}$$

are linearly independent. To see this suppose that $r_1w_1 + r_2w_2 = 0$. Using the components of w_1 and w_2 this equality is equivalent to the system of four equations

$$2r_1 + r_2 = 0, \quad -8r_1 - 2r_2 = 0, \quad r_1 = 0, \quad \text{and} \quad r_2 = 0.$$

In particular, $r_1 = r_2 = 0$; hence w_1 and w_2 are linearly independent.

Using MATLAB to Decide Linear Dependence

Suppose that we want to determine whether or not the vectors

$$w_1 = \begin{pmatrix} 1 \\ 2 \\ -1 \\ 3 \\ 5 \end{pmatrix} \quad w_2 = \begin{pmatrix} -1 \\ 1 \\ 4 \\ -2 \\ 0 \end{pmatrix} \quad w_3 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ 3 \\ 12 \end{pmatrix} \quad w_4 = \begin{pmatrix} 0 \\ 4 \\ 3 \\ 1 \\ -2 \end{pmatrix}$$

(5.4.4*)

are linearly dependent. After typing `e5_4_4` in MATLAB, form the 5×4 matrix A by typing

```
A = [w1 w2 w3 w4]
```

Determine whether there is a nonzero solution to $AR = 0$ by typing

```
null(A)
```

The response from MATLAB is

```
ans =
   -0.7559
   -0.3780
    0.3780
    0.3780
```

showing that there is a nonzero solution to $AR = 0$ and the vectors w_j are linearly dependent. Indeed, this solution for R shows that we can solve for w_1 in terms of w_2, w_3, w_4. We can now ask whether or not w_2, w_3, w_4 are linearly dependent. To answer this question form the matrix

```
B = [w2 w3 w4]
```

and type `null(B)` to obtain

```
ans =
   Empty matrix: 3-by-0
```

showing that the only solution to $BR = 0$ is the zero solution $R = 0$. Thus, w_2, w_3, w_4 are linearly independent. For these particular vectors, any three of the four are linearly independent.

Exercises

5.5 Dimension and Bases

The minimum number of vectors that span a vector space has special significance.

Definition 5.5.1. The vector space V has *finite dimension* if V is the span of a finite number of vectors. If V has finite dimension, then the smallest number of vectors that span V is called the *dimension* of V and is denoted by $\dim V$.

For example, recall that e_j is the vector in $\mathbb{R}^n$ whose j^{th} component is 1 and all of whose other components are 0. Let $x = (x_1, \ldots, x_n)$ be in $\mathbb{R}^n$. Then

$$x = x_1e_1 + \cdots + x_ne_n. \tag{5.5.1}$$

Since every vector in $\mathbb{R}^n$ is a linear combination of the vectors $e_1, \ldots, e_n$, it follows that $\mathbb{R}^n = \mathrm{span}\{e_1, \ldots, e_n\}$. Thus, $\mathbb{R}^n$ is finite dimensional. Moreover, the dimension of $\mathbb{R}^n$ is at most n, since $\mathbb{R}^n$ is spanned by n vectors. It seems unlikely that $\mathbb{R}^n$ could be spanned by fewer than n vectors— but this point needs to be proved.

An Example of a Vector Space that is Not Finite Dimensional Next we discuss an example of a vector space that does not have finite dimension. Consider the subspace $\mathcal{P} \subset \mathcal{C}^1$ consisting of polynomials of all degrees. We show that $\mathcal{P}$ is not the span of a finite number of vectors and hence that $\mathcal{P}$ does not have finite dimension. Let $p_1(t), p_2(t), \ldots, p_k(t)$ be a set of k polynomials and let d be the maximum degree of these k polynomials. Then every polynomial in the span of $p_1(t), \ldots, p_k(t)$ has degree less than or equal to d. In particular, $p(t) = t^{d+1}$ is a polynomial that is not in the span of $p_1(t), \ldots, p_k(t)$ and $\mathcal{P}$ is not spanned by finitely many vectors.

Bases and The Main Theorem

Definition 5.5.2. Let $\mathcal{B} = \{w_1, \ldots, w_k\}$ be a set of vectors in a vector space W. The subset $\mathcal{B}$ is a *basis* for W if $\mathcal{B}$ is a spanning set for W with the smallest number of elements in a spanning set for W.

It follows that if $\{w_1, \ldots, w_k\}$ is a basis for W, then $k = \dim W$. The main theorem about bases is:

Theorem 5.5.3. *A set of vectors $\mathcal{B} = \{w_1, \ldots, w_k\}$ in a vector space W is a basis for W if and only if the set $\mathcal{B}$ is linearly independent and spans W.*

Remark: The importance of Theorem 5.5.3 is that we can show that a set of vectors is a basis by verifying spanning and linear independence. We never have to check directly that the spanning set has the minimum number of vectors for a spanning set.

For example, we have shown previously that the set of vectors $\{e_1, \ldots, e_n\}$ in $\mathbb{R}^n$ is linearly independent and spans $\mathbb{R}^n$. It follows from Theorem 5.5.3 that this set is a basis, and that the dimension of $\mathbb{R}^n$ is n. In particular, $\mathbb{R}^n$ cannot be spanned by fewer than n vectors.

The proof of Theorem 5.5.3 is given in Section 5.6.

Consequences of Theorem 5.5.3 We discuss two applications of Theorem 5.5.3. First, we use this theorem to derive a way of determining the dimension of the subspace spanned by a finite number of vectors. Second, we show that the dimension of the subspace of solutions to a homogeneous system of linear equation $Ax = 0$ is $n - \mathrm{rank}(A)$ where A is an $m \times n$ matrix.

Computing the Dimension of a Span We show that the dimension of a span of vectors can be found using elementary row operations on M.

Lemma 5.5.4. *Let $w_1, \ldots, w_k$ be k row vectors in $\mathbb{R}^n$*

and let $W = \mathrm{span}\{w_1, \ldots, w_k\} \subset \mathbb{R}^n$. *Define*

$$M = \begin{pmatrix} w_1 \\ \vdots \\ w_k \end{pmatrix}$$

to be the matrix whose rows are the w_j*s. Then*

$$\dim(W) = \mathrm{rank}(M). \tag{5.5.2}$$

Proof To verify (5.5.2), observe that the span of $w_1, \ldots, w_k$ is unchanged by

(a) swapping w_i and w_j,

(b) multiplying w_i by a nonzero scalar, and

(c) adding a multiple of w_i to w_j.

That is, if we perform elementary row operations on M, the vector space spanned by the rows of M does not change. So we may perform elementary row operations on M until we arrive at the matrix E in reduced echelon form. Suppose that $\ell = \mathrm{rank}(M)$; that is, suppose that ℓ is the number of nonzero rows in E. Then

$$E = \begin{pmatrix} v_1 \\ \vdots \\ v_\ell \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

where the v_j are the nonzero rows in the reduced echelon form matrix.

We claim that the vectors $v_1, \ldots, v_\ell$ are linearly independent. It then follows from Theorem 5.5.3 that $\{v_1, \ldots, v_\ell\}$ is a basis for W and that the dimension of W is ℓ. To verify the claim, suppose

$$a_1 v_1 + \cdots + a_\ell v_\ell = 0. \tag{5.5.3}$$

We show that a_i must equal 0 as follows. In the i^{th} row, the pivot must occur in some column — say in the j^{th} column. It follows that the j^{th} entry in the vector of the left hand side of (5.5.3) is

$$0a_1 + \cdots + 0a_{i-1} + 1a_i + 0a_{i+1} + \cdots + 0a_\ell = a_i,$$

since all entries in the j^{th} column of E other than the pivot must be zero, as E is in reduced echelon form. ■

For instance, let $W = \mathrm{span}\{w_1, w_2, w_3\}$ in $\mathbb{R}^4$ where

$$\begin{array}{rcl} w_1 & = & (3, -2, 1, -1), \\ w_2 & = & (1, 5, 10, 12), \\ w_3 & = & (1, -12, -19, -25). \end{array} \tag{5.5.4*}$$

To compute $\dim W$ in MATLAB , type `e5_5_4` to load the vectors and type

```
M = [w1; w2; w3]
```

Row reduction of the matrix `M` in MATLAB leads to the reduced echelon form matrix

```
ans =
    1.0000         0    1.4706    1.1176
         0    1.0000    1.7059    2.1765
         0         0         0         0
```

indicating that the dimension of the subspace W is two, and therefore $\{w_1, w_2, w_3\}$ is not a basis of W. Alternatively, we can use the MATLAB command `rank(M)` to compute the rank of M and the dimension of the span W.

However, if we change one of the entries in w_3, for instance `w3(3)=-18` then indeed the command `rank([w1;w2;w3])` gives the answer three indicating that for this choice of vectors $\{w1, w2, w3\}$ is a basis for $\mathrm{span}\{w1, w2, w3\}$.

Solutions to Homogeneous Systems Revisited We return to our discussions in Chapter 2 on solving linear equations. Recall that we can write all solutions to the system of homogeneous equations $Ax = 0$ in terms of a few parameters, and that the null space of A is the subspace of solutions (See Definition 5.2.1). More precisely, Proposition 5.2.5 states that the number of parameters needed is $n - \mathrm{rank}(A)$ where n is the number of variables in the homogeneous system. We claim that the dimension of the null space is exactly $n - \mathrm{rank}(A)$.

For example, consider the reduced echelon form 3×7 matrix

$$A = \begin{pmatrix} 1 & -4 & 0 & 2 & -3 & 0 & 8 \\ 0 & 0 & 1 & 3 & 2 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix} \tag{5.5.5}$$

that has rank three. Suppose that the unknowns for this system of equations are $x_1, \ldots, x_7$. We can solve the equations associated with A by solving the first equation for x_1, the second equation for x_3, and the third equation for x_6, as follows:

$$\begin{aligned} x_1 &= 4x_2 - 2x_4 + 3x_5 - 8x_7 \\ x_3 &= -3x_4 - 2x_5 - 4x_7 \\ x_6 &= -2x_7 \end{aligned}$$

Thus, all solutions to this system of equations have the form

$$\begin{pmatrix} 4x_2 - 2x_4 + 3x_5 - 8x_7 \\ x_2 \\ -3x_4 - 2x_5 - 4x_7 \\ x_4 \\ x_5 \\ -2x_7 \\ x_7 \end{pmatrix} \tag{5.5.6}$$

which equals

$$x_2 \begin{pmatrix} 4 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + x_4 \begin{pmatrix} -2 \\ 0 \\ -3 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + x_5 \begin{pmatrix} 3 \\ 0 \\ -2 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + x_7 \begin{pmatrix} -8 \\ 0 \\ -4 \\ 0 \\ 0 \\ -2 \\ 1 \end{pmatrix}.$$

We can rewrite the right hand side of (5.5.6) as a linear combination of four vectors w_2, w_4, w_5, w_7

$$x_2 w_2 + x_4 w_4 + x_5 w_5 + x_7 w_7. \tag{5.5.7}$$

This calculation shows that the null space of A, which is $W = \{x \in \mathbb{R}^7 : Ax = 0\}$, is spanned by the four vectors w_2, w_4, w_5, w_7. Moreover, this same calculation shows that the four vectors are linearly independent. From the left hand side of (5.5.6) we see that if this linear combination sums to zero, then $x_2 = x_4 = x_5 = x_7 = 0$. It follows from Theorem 5.5.3 that $\dim W = 4$.

Definition 5.5.5. The *nullity* of A is the dimension of the null space of A.

Theorem 5.5.6. *Let A be an $m \times n$ matrix. Then*

$$\mathrm{nullity}(A) + \mathrm{rank}(A) = n.$$

Proof Neither the rank nor the null space of A are changed by elementary row operations. So we can assume that A is in reduced echelon form. The rank of A is the number of nonzero rows in the reduced echelon form matrix. Proposition 5.2.5 states that the null space is spanned by p vectors where $p = n - \mathrm{rank}(A)$. We must show that these vectors are linearly independent.

Let $j_1, \ldots, j_p$ be the columns of A that do not contain pivots. In example (5.5.5) $p = 4$ and

$$j_1 = 2, \qquad j_2 = 4, \qquad j_3 = 5, \qquad j_4 = 7.$$

After solving for the variables corresponding to pivots, we find that the spanning set of the null space consists of p vectors in $\mathbb{R}^n$, which we label as $\{w_{j_1}, \ldots, w_{j_p}\}$. See (5.5.6). Note that the ${j_m}^{th}$ entry of w_{j_m} is 1 while the ${j_m}^{th}$ entry in all of the other $p-1$ vectors is 0. Again, see (5.5.6) as an example that supports this statement. It follows that the set of spanning vectors is a linearly independent set. That is, suppose that

$$r_1 w_{j_1} + \cdots + r_p w_{j_p} = 0.$$

From the ${j_m}^{th}$ entry in this equation, it follows that $r_m = 0$; and the vectors are linearly independent. ■

Theorem 5.5.6 has an interesting and useful interpretation. We have seen in the previous subsection that the rank of a matrix A is just the number of linearly independent rows in A. In linear systems each row of the coefficient matrix corresponds to a linear equation. Thus, the rank of A may be thought of as the number of independent equations in a system of linear equations. This theorem just states that the space of solutions loses a dimension for each independent equation.

Exercises

5.6 The Proof of the Main Theorem

We begin the proof of Theorem 5.5.3 with two lemmas on linearly independent and spanning sets.

Lemma 5.6.1. *Let $\{w_1, \ldots, w_k\}$ be a set of vectors in a vector space V and let W be the subspace spanned by these vectors. Then there is a linearly independent subset of $\{w_1, \ldots, w_k\}$ that also spans W.*

Proof If $\{w_1, \ldots, w_k\}$ is linearly independent, then the lemma is proved. If not, then the set $\{w_1, \ldots, w_k\}$ is linearly dependent. If this set is linearly dependent, then at least one of the vectors is a linear combination of the others. By renumbering if necessary, we can assume that w_k is a linear combination of $w_1, \ldots, w_{k-1}$; that is,

$$w_k = a_1 w_1 + \cdots + a_{k-1} w_{k-1}.$$

Now suppose that $w \in W$. Then

$$w = b_1 w_1 + \cdots + b_k w_k.$$

It follows that

$$w = (b_1 + b_k a_1) w_1 + \cdots + (b_{k-1} + b_k a_{k-1}) w_{k-1},$$

and that $W = \mathrm{span}\{w_1, \ldots, w_{k-1}\}$. If the vectors $w_1, \ldots, w_{k-1}$ are linearly independent, then the proof of the lemma is complete. If not, continue inductively until a linearly independent subset of the w_j that also spans W is found. ■

The important point in proving that linear independence together with spanning imply that we have a basis is discussed in the next lemma.

Lemma 5.6.2. *Let W be an m-dimensional vector space and let $k > m$ be an integer. Then any set of k vectors in W is linearly dependent.*

Proof Since the dimension of W is m we know that this vector space can be written as $W = \mathrm{span}\{v_1, \ldots, v_m\}$. Moreover, Lemma 5.6.1 implies that the vectors $v_1, \ldots, v_m$ are linearly independent. Suppose that $\{w_1, \ldots, w_k\}$ is another set of vectors where $k > m$. We have to show that the vectors $w_1, \ldots, w_k$ are linearly dependent; that is, we must show that there exist scalars $r_1, \ldots, r_k$ not all of which are zero that satisfy

$$r_1 w_1 + \cdots + r_k w_k = 0. \tag{5.6.1}$$

We find these scalars by solving a system of linear equations, as we now show.

The fact that W is spanned by the vectors v_j implies that

$$\begin{aligned}
w_1 &= a_{11} v_1 + \cdots + a_{m1} v_m \\
w_2 &= a_{12} v_1 + \cdots + a_{m2} v_m \\
&\vdots \\
w_k &= a_{1k} v_1 + \cdots + a_{mk} v_m.
\end{aligned}$$

It follows that $r_1 w_1 + \cdots + r_k w_k$ equals

$$\begin{array}{ll}
r_1(a_{11} v_1 + \cdots + a_{m1} v_m) & + \\
r_2(a_{12} v_1 + \cdots + a_{m2} v_m) & + \cdots + \\
r_k(a_{1k} v_1 + \cdots + a_{mk} v_m) &
\end{array}$$

Rearranging terms leads to the expression:

$$\begin{array}{ll}
(a_{11} r_1 + \cdots + a_{1k} r_k) v_1 & + \\
(a_{21} r_1 + \cdots + a_{2k} r_k) v_2 & + \cdots + \\
(a_{m1} r_1 + \cdots + a_{mk} r_k) v_m. &
\end{array} \tag{5.6.2}$$

Thus, (5.6.1) is valid if and only if (5.6.2) sums to zero. Since the set $\{v_1, \ldots, v_m\}$ is linearly independent, (5.6.2) can equal zero if and only if

$$\begin{aligned}
a_{11} r_1 + \cdots + a_{1k} r_k &= 0 \\
a_{21} r_1 + \cdots + a_{2k} r_k &= 0 \\
&\vdots \\
a_{m1} r_1 + \cdots + a_{mk} r_k &= 0.
\end{aligned}$$

Since $m < k$, Chapter 2, Theorem 2.4.6 implies that this system of homogeneous linear equations always has a nonzero solution $r = (r_1, \ldots, r_k)$ — from which it follows that the w_i are linearly dependent. ■

Corollary 5.6.3. *Let V be a vector space of dimension n and let $\{u_1, \ldots, u_k\}$ be a linearly independent set of vectors in V. Then $k \le n$.*

Proof If $k > n$ then Lemma 5.6.2 implies that $\{u_1, \ldots, u_k\}$ is linearly dependent. Since we have assumed that this set is linearly independent, it follows that $k \le n$. ■

Proof of Theorem 5.5.3 Suppose that $\mathcal{B} = \{w_1, \ldots, w_k\}$ is a basis for W. By definition, $\mathcal{B}$ spans W and $k = \dim W$. We must show that $\mathcal{B}$ is linearly independent. Suppose $\mathcal{B}$ is linearly dependent, then Lemma 5.6.1 implies that there is a proper subset of $\mathcal{B}$ that spans W (and is linearly independent). This contradicts the fact that as a basis $\mathcal{B}$ has the smallest number of elements of any spanning set for W.

Suppose that $\mathcal{B} = \{w_1, \ldots, w_k\}$ both spans W and is linearly independent. Linear independence and Corollary 5.6.3 imply that $k \le \dim W$. Since, by definition, any spanning set of W has at least $\dim W$ vectors, it follows that $k \ge \dim W$. Thus, $k = \dim W$ and $\mathcal{B}$ is a basis. ■

Extending Linearly Independent Sets to Bases Lemma 5.6.1 leads to one approach to finding bases. Suppose that the subspace W is spanned by a finite set of vectors $\{w_1, \ldots, w_k\}$. Then, we can throw out vectors one by one until we arrive at a linearly independent subset of the w_j. This subset is a basis for W.

We now discuss a second approach to finding a basis for a nonzero subspace W of a finite dimensional vector space V.

Lemma 5.6.4. *Let $\{u_1, \ldots, u_k\}$ be a linearly independent set of vectors in a vector space V and assume that*

$$u_{k+1} \notin \operatorname{span}\{u_1, \ldots, u_k\}.$$

Then $\{u_1, \ldots, u_{k+1}\}$ is also a linearly independent set.

Proof Let $r_1, \ldots, r_{k+1}$ be scalars such that

$$r_1 u_1 + \cdots + r_{k+1} u_{k+1} = 0. \tag{5.6.3}$$

To prove independence, we need to show that all $r_j = 0$. Suppose $r_{k+1} \neq 0$. Then we can solve (5.6.3) for

$$u_{k+1} = -\frac{1}{r_{k+1}}(r_1 u_1 + \cdots + r_k u_k),$$

which implies that $u_{k+1} \in \operatorname{span}\{u_1, \ldots, u_k\}$. This contradicts the choice of u_{k+1}. So $r_{k+1} = 0$ and

$$r_1 u_1 + \cdots + r_k u_k = 0.$$

Since $\{u_1, \ldots, u_k\}$ is linearly independent, it follows that $r_1 = \cdots = r_k = 0$. ■

The second method for constructing a basis is:

- Choose a nonzero vector w_1 in W.
- If W is not spanned by w_1, then choose a vector w_2 that is not on the line spanned by w_1.
- If $W \neq \operatorname{span}\{w_1, w_2\}$, then choose a vector $w_3 \notin \operatorname{span}\{w_1, w_2\}$.
- If $W \neq \operatorname{span}\{w_1, w_2, w_3\}$, then choose a vector $w_4 \notin \operatorname{span}\{w_1, w_2, w_3\}$.

- Continue until a spanning set for W is found. This set is a basis for W.

We now justify this approach to finding bases for subspaces. Suppose that W is a subspace of a finite dimensional vector space V. For example, suppose that $W \subset \mathbb{R}^n$. Then our approach to finding a basis of W is as follows. Choose a nonzero vector $w_1 \in W$. If $W = \mathrm{span}\{w_1\}$, then we are done. If not, choose a vector $w_2 \in W - \mathrm{span}\{w_1\}$. It follows from Lemma 5.6.4 that $\{w_1, w_2\}$ is linearly independent. If $W = \mathrm{span}\{w_1, w_2\}$, then Theorem 5.5.3 implies that $\{w_1, w_2\}$ is a basis for W, $\dim W = 2$, and we are done. If not, choose $w_3 \in W - \mathrm{span}\{w_1, w_2\}$ and $\{w_1, w_2, w_3\}$ is linearly independent. The finite dimension of V implies that continuing inductively must lead to a spanning set of linear independent vectors for W — which by Theorem 5.5.3 is a basis. This discussion proves:

Corollary 5.6.5. *Every linearly independent subset of a finite dimensional vector space V can be extended to a basis of V.*

Further consequences of Theorem 5.5.3 We summarize here several important facts about dimensions.

Corollary 5.6.6. *Let W be a subspace of a finite dimensional vector space V.*

(a) Suppose that W is a proper subspace. Then $\dim W < \dim V$.

(b) Suppose that $\dim W = \dim V$. Then $W = V$.

Proof (a) Let $\dim W = k$ and let $\{w_1, \ldots, w_k\}$ be a basis for W. Since W is a proper subspace of V, there is a vector $w \in V - W$. It follows from Lemma 5.6.4 that $\{w_1, \ldots, w_k, w\}$ is a linearly independent set. Therefore, Corollary 5.6.3 implies that $k + 1 \leq n$.

(b) Let $\{w_1, \ldots, w_k\}$ be a basis for W. Theorem 5.5.3 implies that this set is linearly independent. If $\{w_1, \ldots, w_k\}$ does not span V, then it can be extended to a basis as above. But then $\dim V > \dim W$, which is a contradiction. ■

Corollary 5.6.7. *Let $\mathcal{B} = \{w_1, \ldots, w_n\}$ be a set of n vectors in an n-dimensional vector space V. Then the following are equivalent:*

(a) $\mathcal{B}$ is a spanning set of V,

(b) $\mathcal{B}$ is a basis for V, and

(c) $\mathcal{B}$ is a linearly independent set.

Proof By definition, (a) implies (b) since a basis is a spanning set with the number of vectors equal to the dimension of the space. Theorem 5.5.3 states that a basis is a linearly independent set; so (b) implies (c). If $\mathcal{B}$ is a linearly independent set of n vectors, then it spans a subspace W of dimension n. It follows from Corollary 5.6.6(b) that $W = V$ and that (c) implies (a). ■

Subspaces of $\mathbb{R}^3$ We can now classify all subspaces of $\mathbb{R}^3$. They are: the origin, lines through the origin, planes through the origin, and $\mathbb{R}^3$. All of these sets were shown to be subspaces in Example 5.1.5(a–c).

To verify that these sets are the only subspaces of $\mathbb{R}^3$, note that Theorem 5.5.3 implies that proper subspaces of $\mathbb{R}^3$ have dimension equal either to one or two. (The zero dimensional subspace is the origin and the only three dimensional subspace is $\mathbb{R}^3$ itself.) One dimensional subspaces of $\mathbb{R}^3$ are spanned by one nonzero vector and are just lines through the origin. See Example 5.1.5(b). We claim that all two dimensional subspaces are planes through the origin.

Suppose that $W \subset \mathbb{R}^3$ is a subspace spanned by two noncollinear vectors w_1 and w_2. We show that W is a plane through the origin using results in Chapter 2. Observe that there is a vector $N = (N_1, N_2, N_3)$ perpendicular to $w_1 = (a_{11}, a_{12}, a_{13})$ and $w_2 = (a_{21}, a_{22}, a_{23})$. Such a vector N satisfies the two linear equations:

$$\begin{aligned} w_1 \cdot N &= a_{11}N_1 + a_{12}N_2 + a_{13}N_3 = 0 \\ w_2 \cdot N &= a_{21}N_1 + a_{22}N_2 + a_{23}N_3 = 0. \end{aligned}$$

Chapter 2, Theorem 2.4.6 implies that a system of two linear equations in three unknowns has a nonzero solution. Let P be the plane perpendicular to N that contains the origin. We show that $W = P$ and hence that the claim is valid.

The choice of N shows that the vectors w_1 and w_2 are both in P. In fact, since P is a subspace it contains every vector in span$\{w_1, w_2\}$. Thus $W \subset P$. If P contains just one additional vector $w_3 \in \mathbb{R}^3$ that is not in W, then the span of w_1, w_2, w_3 is three dimensional and $P = W = \mathbb{R}^3$.

Exercises

6 Closed Form Solutions for Planar ODEs

In this chapter we describe several methods for finding closed form solutions to planar constant coefficient systems of linear differential equations and we use these methods to discuss qualitative features of phase portraits of these solutions.

In Section 6.1 we show how uniqueness to initial value problems implies that the space of solutions to a constant coefficient system of n linear differential equations is n dimensional. Using this observation we present a direct method for solving planar linear systems in Section 6.2. This method extends the discussion of solutions to systems whose coefficient matrices have distinct real eigenvalues given in Section 4.7 to the cases of complex eigenvalues and equal real eigenvalues.

A second method for finding solutions is to use changes of coordinates to make the coefficient matrix of the differential equation as simple as possible. This idea leads to the notion of *similarity* of matrices, which is discussed in Section 6.3, and leads to the second method for solving planar linear systems. Similarity also leads to the Jordan Normal Form theorem for 2×2 matrices. Both the direct method and the method based on similarity require being able to compute the eigenvalues and eigenvectors of the coefficient matrix.

The important subject of qualitative features of phase portraits of linear systems are explored in Section 6.4. Specifically we discuss *saddles*, *sinks*, *sources* and *asymptotic stability*. This discussion also uses similarity and Jordan Normal Form. We find that the qualitative theory is determined by the eigenvalues and eigenvectors of the coefficient matrix — which is not surprising given that we can classify matrices up to similarity by just knowing their eigenvalues and eigenvectors.

Chapter 6 ends with three optional sections. Matrix exponentials yield an elegant third way to derive closed form solutions to n-dimensional linear ODE systems (Section 6.5). This method leads to a proof of uniqueness of solutions to initial value problems of linear systems (Theorem 6.5.1). A proof of the Cayley Hamilton Theorem for 2×2 matrices is given in Section 6.6. In the last section, Section 6.7, we obtain solutions to second order equations by reducting them to first order systems.

6.1 The Initial Value Problem

Recall that a planar autonomous constant coefficient system of ordinary differential equations has the form

$$\begin{aligned} \frac{dx}{dt} &= ax+by \\ \frac{dy}{dt} &= cx+dy \end{aligned} \tag{6.1.1}$$

where $a,b,c,d \in \mathbb{R}$. Computer experiments using pplane10 lead us to believe that there is just one solution to (6.1.1) satisfying the initial conditions

$$\begin{aligned} x(0) &= x_0 \\ y(0) &= y_0. \end{aligned}$$

We prove existence in this section and the next by determining explicit formulas for solutions.

The Initial Value Problem for Linear Systems In this chapter we discuss how to find solutions $(x(t),y(t))$ to (6.1.1) satisfying the initial values $x(0)=x_0$ and $y(0)=y_0$. It is convenient to rewrite (6.1.1) in matrix form as:

$$\frac{dX}{dt}(t)=CX(t). \tag{6.1.2}$$

The initial value problem is then stated as: Find a solution to (6.1.2) satisfying $X(0)=X_0$ where $X_0=(x_0,y_0)^t$. Everything that we have said here works equally well for n dimensional systems of linear differential equations. Just let C be an $n\times n$ matrix and let X_0 be an n vector of initial conditions.

Solving the Initial Value Problem Using Superposition In Section 4.7 we discussed how to solve (6.1.2) when the eigenvalues of C are real and distinct. Recall that when λ_1 and λ_2 are distinct real eigenvalues of C with associated eigenvectors v_1 and v_2, there are two solutions to (6.1.2) given by the explicit formulas

$$X_1(t)=e^{\lambda_1 t}v_1 \quad \text{and} \quad X_2(t)=e^{\lambda_2 t}v_2.$$

Superposition guarantees that every linear combination of these solutions

$$X(t)=\alpha_1 X_1(t)+\alpha_2 X_2(t)=\alpha_1 e^{\lambda_1 t}v_1+\alpha_2 e^{\lambda_2 t}v_2$$

is a solution to (6.1.2). Since v_1 and v_2 are linearly independent, we can always choose scalars $\alpha_1,\alpha_2\in\mathbb{R}$ to solve any given initial value problem of (6.1.2). It follows from the uniqueness of solutions to initial value problems that all solutions to (6.1.2) are included in this family of solutions. Uniqueness is proved in the special case of linear systems in Theorem 6.5.1. This proof uses matrix exponentials.

We generalize this discussion so that we will be able to find closed form solutions to (6.1.2) in Section 6.2 when the eigenvalues of C are complex or are real and equal.

Suppose that $X_1(t)$ and $X_2(t)$ are two solutions to (6.1.1) such that

$$v_1=X_1(0) \quad \text{and} \quad v_2=X_2(0)$$

are linearly independent. Then all solutions to (6.1.1) are linear combinations of these two solutions. We verify this statement as follows. Corollary 5.6.7 of Chapter 5 states that since $\{v_1,v_2\}$ is a linearly independent set in $\mathbb{R}^2$, it is also a basis of $\mathbb{R}^2$. Thus for every $X_0\in\mathbb{R}^2$ there exist scalars r_1,r_2 such that

$$X_0=r_1v_1+r_2v_2.$$

It follows from superposition that the solution

$$X(t)=r_1X_1(t)+r_2X_2(t)$$

is the unique solution whose initial condition vector is X_0.

We have proved that every solution to this linear system of differential equations is a linear combination of these two solutions — that is, we have proved that the dimension of the space of solutions to (6.1.2) is two. This proof generalizes immediately to a proof of the following theorem for $n\times n$ systems.

Theorem 6.1.1. *Let C be an $n \times n$ matrix. Suppose that*

$$X_1(t), \ldots, X_n(t)$$

are solutions to $\dot{X} = CX$ such that the vectors of initial conditions $v_j = X_j(0)$ are linearly independent in $\mathbb{R}^n$. Then the unique solution to the system (6.1.2) with initial condition $X(0) = X_0$ is

$$X(t) = r_1 X_1(t) + \cdots + r_n X_n(t), \tag{6.1.3}$$

where $r_1, \ldots, r_n$ are scalars satisfying

$$X_0 = r_1 v_1 + \cdots + r_n v_n. \tag{6.1.4}$$

We call (6.1.3) the *general solution* to the system of differential equations $\dot{X} = CX$. When solving the initial value problem we find a *particular solution* by specifying the scalars $r_1, \ldots, r_n$.

Corollary 6.1.2. *Let C be an $n \times n$ matrix and let*

$$\mathcal{X} = \{X_1(t), \ldots, X_n(t)\}$$

be solutions to the differential equation $\dot{X} = CX$ such that the vectors $X_j(0)$ are linearly independent in $\mathbb{R}^n$. Then the set of all solutions to $\dot{X} = CX$ is an n-dimensional subspace of $(\mathcal{C}^1)^n$, and $\mathcal{X}$ is a basis for the solution subspace.

Consider a special case of Theorem 6.1.1. Suppose that the matrix C has n linearly independent eigenvectors $v_1, \ldots, v_n$ with real eigenvalues $\lambda_1, \ldots, \lambda_n$. Then the functions $X_j(t) = e^{\lambda_j t} v_j$ are solutions to $\dot{X} = CX$. Corollary 6.1.2 implies that the functions X_j form a basis for the space of solutions of this system of differential equations. Indeed, the general solution to (6.1.2) is

$$X(t) = r_1 e^{\lambda_1 t} v_1 + \cdots + r_n e^{\lambda_n t} v_n. \tag{6.1.5}$$

The particular solution that solves the initial value $X(0) = X_0$ is found by solving (6.1.4) for the scalars $r_1, \ldots, r_n$.

Exercises

6.2 Closed Form Solutions by the Direct Method

In Section 4.7 we showed in detail how solutions to planar systems of constant coefficient differential equations with distinct real eigenvalues are found. This method was just reviewed in Section 6.1 where we saw that the crucial step in solving these systems of differential equations is the step where we find two linearly independent solutions. In this section we discuss how to find these two linearly independent solutions when the eigenvalues of the coefficient matrix are either complex or real and equal.

By finding these two linearly independent solutions we will find both the *general* solution of the system of differential equations $\dot{X} = CX$ and a method for solving the initial value problem

$$\begin{array}{rcl} \dfrac{dX}{dt} & = & CX \\ X(0) & = & X_0. \end{array} \tag{6.2.1}$$

The principle results of this section are summarized as follows. Let C be a 2×2 matrix with eigenvalues λ_1 and λ_2, and associated eigenvectors v_1 and v_2.

(a) If the eigenvalues are real and v_1 and v_2 are linearly independent, then the general solution to (6.2.1) is given by (6.2.2).

(b) If the eigenvalues are complex, then the general solution to (6.2.1) is given by (6.2.3) and (6.2.4).

(c) If the eigenvalues are equal (and hence real) and there is only one linearly independent eigenvector, then the general solution to (6.2.1) is given by (6.2.16).

Real Distinct Eigenvalues We have discussed the case when $\lambda_1 \neq \lambda_2 \in \mathbb{R}$ on several occasions. For completeness we repeat the result. The general solution is:

$$X(t) = \alpha_1 e^{\lambda_1 t} v_1 + \alpha_2 e^{\lambda_2 t} v_2. \tag{6.2.2}$$

The initial value problem is solved by finding real numbers α_1 and α_2 such that

$$X_0 = \alpha_1 v_1 + \alpha_2 v_2.$$

See Section 4.7 for a detailed discussion with examples.

Complex Conjugate Eigenvalues Suppose that the eigenvalues of C are complex, that is, suppose that $\lambda_1 = \sigma + i\tau$ with $\tau \neq 0$ is an eigenvalue of C with eigenvector $v_1 = v + iw$, where $v, w \in \mathbb{R}^2$. We claim that $X_1(t)$ and $X_2(t)$, where

$$\begin{array}{rcl} X_1(t) & = & e^{\sigma t}(\cos(\tau t)v - \sin(\tau t)w) \\ X_2(t) & = & e^{\sigma t}(\sin(\tau t)v + \cos(\tau t)w), \end{array} \tag{6.2.3}$$

are solutions to (6.2.1) and that the general solution to (6.2.1) is:

$$X(t) = \alpha_1 X_1(t) + \alpha_2 X_2(t), \tag{6.2.4}$$

where α_1, α_2 are real scalars.

There are several difficulties in deriving (6.2.3) and (6.2.4); these difficulties are related to using complex numbers as opposed to real numbers. In particular, in the derivation of (6.2.3) we need to define the exponential of a complex number, and we begin by discussing this issue.

Euler's Formula We find complex exponentials by using Euler's celebrated formula:

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{6.2.5}$$

for any real number θ. A justification of this formula is given in Exercise **??**. Euler's formula allows us to differentiate complex exponentials, obtaining the expected result:

$$\begin{aligned}\frac{d}{dt}e^{i\tau t} &= \frac{d}{dt}(\cos(\tau t)+i\sin(\tau t))\\ &= \tau(-\sin(\tau t)+i\cos(\tau t))\\ &= i\tau(\cos(\tau t)+i\sin(\tau t))\\ &= i\tau e^{i\tau t}.\end{aligned}$$

Euler's formula also implies that

$$e^{\lambda t}=e^{\sigma t+i\tau t}=e^{\sigma t}e^{i\tau t}=e^{\sigma t}(\cos(\tau t)+i\sin(\tau t)), \quad (6.2.6)$$

where $\lambda=\sigma+i\tau$. Most importantly, we note that

$$\frac{d}{dt}e^{\lambda t}=\lambda e^{\lambda t}. \quad (6.2.7)$$

We use (6.2.6) and the product rule for differentiation to verify (6.2.7) as follows:

$$\begin{aligned}\frac{d}{dt}e^{\lambda t} &= \frac{d}{dt}\left(e^{\sigma t}e^{i\tau t}\right)\\ &= \left(\sigma e^{\sigma t}\right)e^{i\tau t}+e^{\sigma t}\left(i\tau e^{i\tau t}\right)\\ &= (\sigma+i\tau)e^{\sigma t+i\tau t}\\ &= \lambda e^{\lambda t}.\end{aligned}$$

Verification that (6.2.4) is the General Solution A complex vector-valued function $X(t)=X_1(t)+iX_2(t)\in\mathbb{C}^n$ consists of a *real part* $X_1(t)\in\mathbb{R}^n$ and an *imaginary part* $X_2(t)\in\mathbb{R}^n$. For such functions $X(t)$ we define

$$\dot{X}=\dot{X}_1+i\dot{X}_2$$

and

$$CX=CX_1+iCX_2.$$

To say that $X(t)$ is a solution to $\dot{X}=CX$ means that

$$\dot{X}_1+i\dot{X}_2=\dot{X}=CX=CX_1+iCX_2. \quad (6.2.8)$$

Lemma 6.2.1. *The complex vector-valued function $X(t)$ is a solution to $\dot{X}=CX$ if and only if the real and imaginary parts are real vector-valued solutions to $\dot{X}=CX$.*

Proof Equating the real and imaginary parts of (6.2.8) implies that $\dot{X}_1=CX_1$ and $\dot{X}_2=CX_2$. ■

It follows from Lemma 6.2.1 that finding one complex-valued solution to a linear differential equation provides us with two real-valued solutions. Identity (6.2.7) implies that

$$X(t)=e^{\lambda_1 t}v_1$$

is a complex-valued solution to (6.2.1). Using Euler's formula we compute the real and imaginary parts of $X(t)$, as follows.

$$\begin{aligned}X(t) &= e^{(\sigma+i\tau)t}(v+iw)\\ &= e^{\sigma t}(\cos(\tau t)+i\sin(\tau t))(v+iw)\\ &= e^{\sigma t}(\cos(\tau t)v-\sin(\tau t)w)\\ &\quad +ie^{\sigma t}(\sin(\tau t)v+\cos(\tau t)w).\end{aligned}$$

Since the real and imaginary parts of $X(t)$ are solutions to $\dot{X}=CX$, it follows that the real-valued functions $X_1(t)$ and $X_2(t)$ defined in (6.2.3) are indeed solutions.

Returning to the case where C is a 2×2 matrix, we see that if $X_1(0)=v$ and $X_2(0)=w$ are linearly independent, then Corollary 6.1.2 implies that (6.2.4) is the general solution to $\dot{X}=CX$. The linear independence of v and w is verified using the following lemma.

Lemma 6.2.2. *Let $\lambda_1=\sigma+i\tau$ with $\tau\neq 0$ be a complex eigenvalue of the 2×2 matrix C with eigenvector $v_1=v+iw$ where $v,w\in\mathbb{R}^2$. Then*

$$\begin{aligned}Cv &= \sigma v-\tau w\\ Cw &= \tau v+\sigma w.\end{aligned} \quad (6.2.9)$$

and v and w are linearly independent vectors.

Proof By assumption $Cv_1 = \lambda_1 v_1$, that is,

$$\begin{array}{rcl} C(v+iw) & = & (\sigma+i\tau)(v+iw) \\ & = & (\sigma v-\tau w)+i(\tau v+\sigma w). \end{array} \tag{6.2.10}$$

Equating real and imaginary parts of (6.2.10) leads to the system of equations (6.2.9). Note that if $w = 0$, then $v \neq 0$ and $\tau v = 0$. Hence $\tau = 0$, contradicting the assumption that $\tau \neq 0$. So $w \neq 0$.

Note also that if v and w are linearly dependent, then $v = \alpha w$. It then follows from the previous equation that

$$Cw = (\tau\alpha + \sigma)w.$$

Hence w is a real eigenvector; but the eigenvalues of C are not real and C has no real eigenvectors. ■

An Example with Complex Eigenvalues Consider an example of an initial value problem for a linear system with complex eigenvalues. Let

$$\frac{dX}{dt} = \begin{pmatrix} -1 & 2 \\ -5 & -3 \end{pmatrix} X = CX, \tag{6.2.11}$$

and

$$X_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The characteristic polynomial for the matrix C is:

$$p_C(\lambda) = \lambda^2 + 4\lambda + 13,$$

whose roots are $\lambda_1 = -2 + 3i$ and $\lambda_2 = -2 - 3i$. So

$$\sigma = -2 \quad \text{and} \quad \tau = 3.$$

An eigenvector corresponding to the eigenvalue λ_1 is

$$v_1 = \begin{pmatrix} 2 \\ -1+3i \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} + i \begin{pmatrix} 0 \\ 3 \end{pmatrix} = v + iw.$$

It follows from (6.2.3) that

$$\begin{array}{rcl} X_1(t) & = & e^{-2t}(\cos(3t)v - \sin(3t)w) \\ X_2(t) & = & e^{-2t}(\sin(3t)v + \cos(3t)w), \end{array}$$

are solutions to (6.2.11) and $X = \alpha_1 X_1 + \alpha_2 X_2$ is the general solution to (6.2.11). To solve the initial value problem we need to find α_1, α_2 such that

$$X_0 = X(0) = \alpha_1 X_1(0) + \alpha_2 X_2(0) = \alpha_1 v + \alpha_2 w,$$

that is,

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \alpha_1 \begin{pmatrix} 2 \\ -1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0 \\ 3 \end{pmatrix}.$$

Therefore, $\alpha_1 = \frac{1}{2}$ and $\alpha_2 = \frac{1}{2}$ and

$$X(t) = e^{-2t} \begin{pmatrix} \cos(3t) + \sin(3t) \\ \cos(3t) - 2\sin(3t) \end{pmatrix}. \tag{6.2.12}$$

Real and Equal Eigenvalues There are two types of 2×2 matrices that have real and equal eigenvalues — those that are scalar multiples of the identity and those that are not. An example of a 2×2 matrix that has real and equal eigenvalues is

$$A = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix}, \quad \lambda_1 \in \mathbb{R}. \tag{6.2.13}$$

The characteristic polynomial of A is

$$p_A(\lambda) = \lambda^2 - \mathrm{tr}(A)\lambda + \det(A) = \lambda^2 - 2\lambda_1\lambda + \lambda_1^2 = (\lambda - \lambda_1)^2.$$

Thus the eigenvalues of A both equal λ_1.

Only One Linearly Independent Eigenvector An important fact about the matrix A in (6.2.13) is that it has only one linearly independent eigenvector. To verify this fact, solve the system of linear equations

$$Av = \lambda_1 v.$$

In matrix form this equation is

$$0 = (A - \lambda_1 I_2)v = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} v.$$

A quick calculation shows that all solutions are multiples of $v_1 = e_1 = (1,0)^t$.

In fact, this observation is valid for any 2×2 matrix that has equal eigenvalues and is not a scalar multiple of the identity, as the next lemma shows.

Lemma 6.2.3. *Let C be a 2×2 matrix. Suppose that C has two linearly independent eigenvectors both with eigenvalue λ_1. Then $C = \lambda_1 I_2$.*

Proof Let v_1 and v_2 be two linearly independent eigenvectors of C; that is, $Cv_j = \lambda_1 v_j$. Since $\dim(\mathbb{R}^2) = 2$, Corollary 5.6.7 implies that $\{v_1, v_2\}$ is a basis of $\mathbb{R}^2$. Hence, every vector v has the form $v = \alpha_1 v_1 + \alpha_2 v_2$. Linearity implies

$$Cv = C(\alpha_1 v_1 + \alpha_2 v_2) = \alpha_1 \lambda_1 v_1 + \alpha_2 \lambda_1 v_2 = \lambda_1 v$$

Therefore, $Cv = \lambda_1 v$ for every $v \in \mathbb{R}^2$ and hence $C = \lambda_1 I_2$. ■

Generalized Eigenvectors Suppose that C has exactly one linearly independent real eigenvector v_1 with a double real eigenvalue λ_1. We call w_1 a *generalized eigenvector* of C it satisfies the system of linear equations

$$(C - \lambda_1 I_2)w_1 = v_1. \tag{6.2.14}$$

The matrix A in (6.2.13) has a generalized eigenvector. To verify this point solve the linear system

$$(C - \lambda_1 I_2)w_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} w_1 = v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

for $w_1 = e_2$. Note that for this matrix C, $v_1 = e_1$ and $w_1 = e_2$ are linearly independent. The next lemma shows that this observation about generalized eigenvectors is always valid.

Lemma 6.2.4. *Let C be a 2×2 matrix with both eigenvalues equal to λ_1 and with one linearly independent eigenvector v_1. Let w_1 be a generalized eigenvector of C, then v_1 and w_1 are linearly independent.*

Proof If v_1 and w_1 were linearly dependent, then w_1 would be a multiple of v_1 and hence an eigenvector of C. But $C - \lambda_1 I_2$ applied to an eigenvector is zero, which is a contradiction. Therefore, v_1 and w_1 are linearly independent. ■

The Cayley Hamilton theorem (see Section 6.6) coupled with matrix exponentials (see Section 6.5) lead to a simple method for finding solutions to differential equations in the multiple eigenvalue case — one that does not require solving for either the eigenvector v_1 or the generalized eigenvector w_1. We next prove the special case of Cayley-Hamilton that is needed.

Lemma 6.2.5. *Let C be a 2×2 matrix with a double eigenvalue $\lambda_1 \in \mathbb{R}$. Then*

$$(C - \lambda_1 I_2)^2 = 0. \tag{6.2.15}$$

Proof Suppose that C has two linearly independent eigenvectors. Then Lemma 6.2.3 implies that $C - \lambda_1 I_2 = 0$ and hence that $(C - \lambda_1 I_2)^2 = 0$.

Suppose that C has one linearly independent eigenvector v_1 and a generalized eigenvector w_1. It follows from Lemma 6.2.4(a) that $\{v_1, w_1\}$ is a basis of $\mathbb{R}^2$. It also follows by definition of eigenvector and generalized eigenvector that

$$\begin{array}{rclcl} (C - \lambda_1 I_2)^2 v_1 & = & (C - \lambda_1 I_2)0 & = & 0 \\ (C - \lambda_1 I_2)^2 w_1 & = & (C - \lambda_1 I_2)v_1 & = & 0 \end{array}$$

Hence, (6.2.15) is valid. ■

Independent Solutions to Differential Equations with Equal Eigenvalues Suppose that the 2×2 matric C has a double eigenvalue λ_1. Then the general solution to the initial value problem $\dot{X}=CX$ and $X(0)=X_0$ is:

$$X(t)=e^{\lambda_1 t}[I_2+t(C-\lambda_1 I_2)]X_0. \tag{6.2.16}$$

This is the form of the solution that is given by matrix exponentials. We verify (6.2.16) by observing that $X(0)=X_0$ and calculating

$$\begin{aligned} CX(t) &= e^{\lambda_1 t}[C+t(C^2-\lambda_1 C)]X_0 \\ \dot{X}(t) &= e^{\lambda_1 t}[\lambda_1(I_2+t(C-\lambda_1 I_2))+(C-\lambda_1 I_2)]X_0. \end{aligned}$$

Therefore

$$CX-\dot{X}=e^{\lambda_1 t}MX_0$$

where (6.2.15) implies

$$\begin{aligned} M &= C+t(C^2-\lambda_1 C)-\lambda_1(I_2+t(C-\lambda_1 I_2)) \\ &\quad -(C-\lambda_1 I_2) \\ &= t(C-\lambda_1 I_2)^2 \\ &= 0. \end{aligned}$$

on use of (6.2.15). A remarkable feature of formula (6.2.16) is that it is not necessary to compute either the eigenvector of C or its generalized eigenvector.

An Example with Equal Eigenvalues Consider the system of differential equations

$$\frac{dX}{dt}=\begin{pmatrix} 1 & -1 \\ 9 & -5 \end{pmatrix} X \tag{6.2.17}$$

with initial value

$$X_0=\begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

The characteristic polynomial for the matrix $\begin{pmatrix} 1 & -1 \\ 9 & -5 \end{pmatrix}$ is

$$p_C(\lambda)=\lambda^2+4\lambda+4=(\lambda+2)^2.$$

Thus $\lambda_1=-2$ is an eigenvalue of multiplicity two. It follows that

$$C-\lambda_1 I_2=\begin{pmatrix} 3 & -1 \\ 9 & -3 \end{pmatrix}$$

and from (6.2.16) that

$$X(t)=e^{-2t}\begin{pmatrix} 1+3t & -t \\ 9t & 1-3t \end{pmatrix}\begin{pmatrix} 2 \\ 3 \end{pmatrix}=e^{-2t}\begin{pmatrix} 2+3t \\ 3+9t \end{pmatrix}.$$

Exercises

6.3 Similar Matrices and Jordan Normal Form

In a certain sense every 2×2 matrix can be thought of as a member of one of three families of matrices. Specifically we show that every 2×2 matrix is similar to one of the matrices listed in Theorem 6.3.4, where similarity is defined as follows.

Definition 6.3.1. The $n \times n$ matrices B and C are *similar* if there exists an invertible $n \times n$ matrix P such that

$$C = P^{-1}BP.$$

Our interest in similar matrices stems from the fact that if we know the solutions to the system of differential equations $\dot{Y} = CY$, then we also know the solutions to the system of differential equations $\dot{X} = BX$. More precisely,

Lemma 6.3.2. *Suppose that B and $C = P^{-1}BP$ are similar matrices. If $Y(t)$ is a solution to the system of differential equations $\dot{Y} = CY$, then $X(t) = PY(t)$ is a solution to the system of differential equations $\dot{X} = BX$.*

Proof Since the entries in the matrix P are constants, it follows that

$$\frac{dX}{dt} = P\frac{dY}{dt}.$$

Since $Y(t)$ is a solution to the $\dot{Y} = CY$ equation, it follows that

$$\frac{dX}{dt} = PCY.$$

Since $Y = P^{-1}X$ and $PCP^{-1} = B$,

$$\frac{dX}{dt} = PCP^{-1}X = BX.$$

Thus $X(t)$ is a solution to $\dot{X} = BX$, as claimed. ■

Invariants of Similarity

Lemma 6.3.3. *Let A and B be similar 2×2 matrices. Then*

$$\begin{array}{rcl} p_A(\lambda) & = & p_B(\lambda), \\ \det(A) & = & \det(B), \\ \mathrm{tr}(A) & = & \mathrm{tr}(B), \end{array}$$

and the eigenvalues of A and B are equal.

Proof The determinant is a function on 2×2 matrices that has several important properties. Recall, in particular, from Chapter 3, Theorem 3.8.2 that for any pair of 2×2 matrices A and B:

$$\det(AB) = \det(A)\det(B), \tag{6.3.1}$$

and for any invertible 2×2 matrix P

$$\det(P^{-1}) = \frac{1}{\det(P)}. \tag{6.3.2}$$

Let P be an invertible 2×2 matrix so that $B = P^{-1}AP$. Using (6.3.1) and (6.3.2) we see that

$$\begin{array}{rcl} p_B(\lambda) & = & \det(B - \lambda I_2) \\ & = & \det(P^{-1}AP - \lambda I_2) \\ & = & \det(P^{-1}(A - \lambda I_2)P) \\ & = & \det(A - \lambda I_2) \\ & = & p_A(\lambda). \end{array}$$

Hence the eigenvalues of A and B are the same. It follows from (4.6.8) and (4.6.9) of Section 4.6 that the determinants and traces of A and B are equal. ■

For example, if

$$A = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix},$$

then

$$P^{-1} = \begin{pmatrix} -1 & 2 \\ 1 & -1 \end{pmatrix}$$

and

$$P^{-1}AP = \begin{pmatrix} 3 & 4 \\ -2 & -3 \end{pmatrix}.$$

A calculation shows that

$$\det(P^{-1}AP) = -1 = \det(A) \quad \text{and} \quad \mathrm{tr}(P^{-1}AP) = 0 = \mathrm{tr}(A),$$

as stated in Lemma 6.3.3.

Classification of Jordan Normal Form 2×2 Matrices We now classify all 2×2 matrices up to similarity.

Theorem 6.3.4. *Let C and $P = (v_1|v_2)$ be 2×2 matrices where the vectors v_1 and v_2 are specified below.*

(a) Suppose that C has two linearly independent real eigenvectors v_1 and v_2 with real eigenvalues λ_1 and λ_2. Then

$$P^{-1}CP = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

(b) Suppose that C has no real eigenvectors and complex conjugate eigenvalues $\sigma \pm i\tau$ where $\tau \neq 0$. Then

$$P^{-1}CP = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix},$$

where $v_1 + iv_2$ is an eigenvector of C associated with the eigenvalue $\lambda_1 = \sigma - i\tau$.

(c) Suppose that C has exactly one linearly independent real eigenvector v_1 with real eigenvalue λ_1. Then

$$P^{-1}CP = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix},$$

where v_2 is a generalized eigenvector of C that satisfies

$$(C - \lambda_1 I_2)v_2 = v_1. \tag{6.3.3}$$

Proof The strategy in the proof of this theorem is to determine the 1^{st} and 2^{nd} columns of $P^{-1}CP$ by computing (in each case) $P^{-1}CPe_j$ for $j = 1$ and $j = 2$. Note from the definition of P that

$$Pe_1 = v_1 \quad \text{and} \quad Pe_2 = v_2.$$

In addition, if P is invertible, then

$$P^{-1}v_1 = e_1 \quad \text{and} \quad P^{-1}v_2 = e_2.$$

Note that if v_1 and v_2 are linearly independent, then P is invertible.

(a) Since v_1 and v_2 are assumed to be linearly independent, P is invertible. So we can compute

$$P^{-1}CPe_1 = P^{-1}Cv_1 = \lambda P^{-1}v_1 = \lambda e_1.$$

It follows that the 1^{st} column of $P^{-1}CP$ is

$$\begin{pmatrix} \lambda_1 \\ 0 \end{pmatrix}.$$

Similarly, the 2^{nd} column of $P^{-1}CP$ is

$$\begin{pmatrix} 0 \\ \lambda_2 \end{pmatrix}$$

thus verifying (a).

(b) Lemma 6.2.2 implies that v_1 and v_2 are linearly independent and hence that P is invertible. Using (6.2.9), with τ replaced by $-\tau$, v replaced by v_1, and w replaced by w_1, we calculate

$$P^{-1}CPe_1 = P^{-1}Cv_1 = \sigma P^{-1}v_1 + \tau P^{-1}v_2 = \sigma e_1 + \tau e_2,$$

and

$$P^{-1}CPe_2 = P^{-1}Cv_2 = -\tau P^{-1}v_1 + \sigma P^{-1}v_2 = -\tau e_1 + \sigma e_2.$$

Thus the columns of $P^{-1}CP$ are

$$\begin{pmatrix} \sigma \\ \tau \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -\tau \\ \sigma \end{pmatrix},$$

as desired.

(c) Let v_1 be an eigenvector and assume that v_2 is a generalized eigenvector satisfying (6.3.3). By Lemma 6.2.4 the vectors v_1 and v_2 exist and are linearly independent.

For this choice of v_1 and v_2, compute

$$P^{-1}CPe_1 = P^{-1}Cv_1 = \lambda_1 P^{-1}v_1 = \lambda_1 e_1,$$

and

$$P^{-1}CPe_2 = P^{-1}Cv_2 = P^{-1}v_1 + \lambda_1 P^{-1}v_2 = e_1 + \lambda_1 e_2.$$

Thus the two columns of $P^{-1}CP$ are:

$$\begin{pmatrix} \lambda_1 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 \\ \lambda_1 \end{pmatrix}.$$

■

Solutions of Jordan Normal Form Equations The eigenvectors of the matrices in Table 2(a) are $v_1 = (1,0)^t$ and $v_2 = (0,1)^t$. Hence, the closed form solution of (a) in that table follows from the direct solution in (6.2.2).

The eigenvectors of the matrices in Table 2(b) are $v_1 = v + iw$ and $v_2 = v - iw$, where $v = (0,1)^t$ and $w = (1,0)^t$. Hence, the closed form solution of (a) in that table follows from the direct solution in (6.2.16)

Finally, the eigenvector and generalized eigenvector of the matrices in Table 2(c) are $v_1 = (1,0)^t$ and $w_1 = (0,1)^t$. Hence, the closed form solution of (c) in that table follows from the direct solution in (6.2.3)

Closed Form Solutions Using Similarity We now use Lemma 6.3.2, Theorem 6.3.4, and the explicit solutions to the normal form equations Table 2 to find solutions for $\dot{X} = CX$ where C is any 2×2 matrix. The idea behind the use of similarity to solve systems of ODEs is to transform a given system into another normal form system whose solution is already known. This method is very much like the technique of change of variables used when finding indefinite integrals in calculus.

We suppose that we are given a system of differential equations $\dot{X} = CX$ and use Theorem 6.3.4 to transform C by similarity to one of the normal form matrices listed in that theorem. We then solve the transformed equation (see Table 2) and use Lemma 6.3.2 to transform the solution back to the given system.

For example, suppose that C has a complex eigenvalue $\sigma - i\tau$ with corresponding eigenvector $v + iw$. Then Theorem 6.3.4 states that

$$B = P^{-1}CP = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix},$$

where $P = (v|w)$ is an invertible matrix. Using Table 2 the general solution to the system of equations $\dot{Y} = BY$ is:

$$Y(t) = e^{\sigma t} \begin{pmatrix} \cos(\tau t) & -\sin(\tau t) \\ \sin(\tau t) & \cos(\tau t) \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

Lemma 6.3.2 states that

$$X(t) = PY(t)$$

is the general solution to the $\dot{X} = CX$ system. Moreover, we can solve the initial value problem by solving

$$X_0 = PY(0) = P \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

for α and β. In particular,

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = P^{-1}X_0.$$

Putting these steps together implies that

$$X(t) = e^{\sigma t} P \begin{pmatrix} \cos(\tau t) & -\sin(\tau t) \\ \sin(\tau t) & \cos(\tau t) \end{pmatrix} P^{-1}X_0 \qquad (6.3.4)$$

is the solution to the initial value problem.

name	normal form equations	closed form solution
(a)	$\dot{X} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} X$	$X(t) = \begin{pmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{pmatrix} X_0$
(b)	$\dot{X} = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix} X$	$X(t) = e^{\sigma t} \begin{pmatrix} \cos(\tau t) & -\sin(\tau t) \\ \sin(\tau t) & \cos(\tau t) \end{pmatrix} X_0$
(c)	$\dot{X} = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix} X$	$X(t) = e^{\lambda_1 t} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} X_0$

Table 2: Solutions to Jordan normal form ODEs with $X(0) = X_0$.

The Example with Complex Eigenvalues Revisited Recall the example in (6.2.11)

$$\frac{dX}{dt} = \begin{pmatrix} -1 & 2 \\ -5 & -3 \end{pmatrix} X,$$

with initial values

$$X_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

This linear system has a complex eigenvalue $\sigma - i\tau = -2 - 3i$ with corresponding eigenvector

$$v + iw = \begin{pmatrix} 2 \\ -1 - 3i \end{pmatrix}.$$

Thus the matrix P that transforms C into normal form is

$$P = \begin{pmatrix} 2 & 0 \\ -1 & -3 \end{pmatrix} \quad \text{and} \quad P^{-1} = \frac{1}{6} \begin{pmatrix} 3 & 0 \\ -1 & -2 \end{pmatrix}.$$

It follows from (6.3.4) that the solution to the initial value problem is

$$\begin{aligned} X(t) &= e^{-2t} P \begin{pmatrix} \cos(3t) & -\sin(3t) \\ \sin(3t) & \cos(3t) \end{pmatrix} P^{-1} X_0 \\ &= \frac{1}{6} e^{-2t} \begin{pmatrix} 2 & 0 \\ -1 & -3 \end{pmatrix} \begin{pmatrix} \cos(3t) & -\sin(3t) \\ \sin(3t) & \cos(3t) \end{pmatrix} \begin{pmatrix} 3 & 0 \\ -1 & -2 \end{pmatrix} X_0. \end{aligned}$$

A calculation gives

$$\begin{aligned} X(t) &= \frac{1}{2} e^{-2t} \begin{pmatrix} 2 & 0 \\ -1 & -3 \end{pmatrix} \begin{pmatrix} \cos(3t) & -\sin(3t) \\ \sin(3t) & \cos(3t) \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ &= e^{-2t} \begin{pmatrix} \cos(3t) + \sin(3t) \\ \cos(3t) - 2\sin(3t) \end{pmatrix}. \end{aligned}$$

Thus the solution to (6.2.11) that we have found using similarity of matrices is identical to the solution (6.2.12) that we found by the direct method.

Solving systems with either distinct real eigenvalues or equal eigenvalues works in a similar fashion.

Exercises

6.4 Sinks, Saddles, and Sources

The qualitative theory of autonomous differential equations begins with the observation that many important properties of solutions to constant coefficient systems of differential equations

$$\frac{dX}{dt} = CX \tag{6.4.1}$$

are unchanged by similarity.

We call the origin of the linear system (6.4.1) a *sink* (or *asymptotically stable*) if all solutions $X(t)$ satisfy

$$\lim_{t\to\infty} X(t) = 0.$$

The origin is a *source* if all nonzero solutions $X(t)$ satisfy

$$\lim_{t\to\infty} ||X(t)|| = \infty.$$

Finally, the origin is a *saddle* if some solutions limit to 0 and some solutions grow infinitely large. Recall also from Lemma 6.3.2 that if $B = P^{-1}CP$, then $P^{-1}X(t)$ is a solution to $\dot{X} = BX$ whenever $X(t)$ is a solution to (6.4.1). Since P^{-1} is a matrix of constants that do not depend on t, it follows that

$$\lim_{t\to\infty} X(t) = 0 \Longleftrightarrow \lim_{t\to\infty} P^{-1}X(t) = 0.$$

or

$$\lim_{t\to\infty} ||X(t)|| = \infty \Longleftrightarrow \lim_{t\to\infty} ||P^{-1}X(t)|| = \infty.$$

It follows the origin is C is a *sink* (or *saddle* or *source*) for (6.4.1) if and only if $P^{-1}X(t)$ is a sink (or saddle or source) for $\dot{X} = BX$.

Theorem 6.4.1. *Consider the system (6.4.1) where C is a 2×2 matrix.*

(a) If the eigenvalues of C have negative real part, then the origin is a sink.

(b) If the eigenvalues of C have positive real part, then the origin is a source.

(c) If one eigenvalue of C is positive and one is negative, then the origin is a saddle.

Proof Lemma 6.3.3 states that the similar matrices B and C have the same eigenvalues. Moreover, as noted the origin is a sink, saddle, or source for B if and only if it is a sink, saddle, or source for C. Thus, we need only verify the theorem for normal form matrices as given in Table 2.

(a) If the eigenvalues λ_1 and λ_2 are real and there are two independent eigenvectors, then Chapter 6, Theorem 6.3.4 states that the matrix C is similar to the diagonal matrix

$$B = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

The general solution to the differential equation $\dot{X} = BX$ is

$$x_1(t) = \alpha_1 e^{\lambda_1 t} \quad \text{and} \quad x_2(t) = \alpha_2 e^{\lambda_2 t}.$$

Since

$$\lim_{t\to\infty} e^{\lambda_1 t} = 0 = \lim_{t\to\infty} e^{\lambda_2 t},$$

when λ_1 and λ_2 are negative, it follows that

$$\lim_{t\to\infty} X(t) = 0$$

for all solutions $X(t)$, and the origin is a sink. Note that if both of the eigenvalues are positive, then $X(t)$ will undergo exponential growth and the origin is a source.

(b) If the eigenvalues of C are the complex conjugates $\sigma \pm i\tau$ where $\tau \neq 0$, then Chapter 6, Theorem 6.3.4 states that after a similarity transformation (6.4.1) has the form

$$\dot{X} = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix} X,$$

and solutions for this equation have the form (6.3.4) of Chapter 6, that is,

$$X(t) = e^{\sigma t}\begin{pmatrix} \cos(\tau t) & -\sin(\tau t) \\ \sin(\tau t) & \cos(\tau t) \end{pmatrix} X_0 = e^{\sigma t} R_{\tau t} X_0,$$

where $R_{\tau t}$ is a rotation matrix (recall (3.2.2) of Chapter 3). It follows that as time evolves the vector X_0 is rotated about the origin and then expanded or contracted by the factor $e^{\sigma t}$. So when $\sigma < 0$, $\lim_{t\to\infty} X(t) = 0$ for all solutions $X(t)$. Hence the origin is a sink and when $\sigma > 0$ solutions spiral away from the origin and the origin is a source.

(c) If the eigenvalues are both equal to λ_1 and if there is only one independent eigenvector, then Chapter 6, Theorem 6.3.4 states that after a similarity transformation (6.4.1) has the form

$$\dot{X} = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix} X,$$

whose solutions are

$$X(t) = e^{t\lambda}\begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} X_0$$

using Table 2(c). Note that the functions $e^{\lambda_1 t}$ and $te^{\lambda_1 t}$ both have limits equal to zero as $t \to \infty$. In the second case, use l'Hôspital's rule and the assumption that $-\lambda_1 > 0$ to compute

$$\lim_{t\to\infty} \frac{t}{e^{-\lambda_1 t}} = -\lim_{t\to\infty} \frac{1}{\lambda_1 e^{-\lambda_1 t}} = 0.$$

Hence $\lim_{t\to\infty} X(t) = 0$ for all solutions $X(t)$ and the origin is asymptotically stable. Note that initially $||X(t)||$ can grow since t is increasing. But eventually exponential decay wins out and solutions limit on the origin. Note that solutions grow exponentially when $\lambda_1 > 0$. ■

Theorem 6.4.1 shows that the qualitative features of the origin for (6.4.1) depend only on the eigenvalues of C and not on the formulae for solutions to (6.4.1). This is a much simpler calculation. However, Theorem 6.4.2 simplifies the calculation substantially further.

Theorem 6.4.2. *(a) If* $\det(C) < 0$, *then* 0 *is a saddle.*

(b) If $\det(C) > 0$ *and* $\mathrm{tr}(C) < 0$, *then* 0 *is a sink.*

(c) If $\det(C) > 0$ *and* $\mathrm{tr}(C) > 0$, *then* 0 *is a source.*

Proof Recall from (4.6.9) that $\det(C)$ is the product of the eigenvalues of C. Hence, if $\det(C) < 0$, then the signs of the eigenvalues must be opposite, and we have a saddle. Next, suppose $\det(C) > 0$. If the eigenvalues are real, then the eigenvalues are either both positive (a source) or both negative (a sink). Recall from (4.6.8) that $\mathrm{tr}(C)$ is sum of the eigenvalues and the sign of the trace determines the sign of the eigenvalues. Finally, assume the eigenvalues are complex conjugates $\sigma \pm i\tau$. Then $\det(C) = \sigma^2 + \tau^2 > 0$ and $\mathrm{tr}(C) = 2\sigma$. Thus, the sign of the real parts of the complex eigenvalues is given by the sign of $\mathrm{tr}(C)$. ■

Time Series It is instructive to note how the time series $x_1(t)$ damps down to the origin in the three cases listed in Theorem 6.4.1. In Figure 18 we present the time series for the three coefficient matrices:

$$C_1 = \begin{pmatrix} -2 & 0 \\ 0 & -1 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} -1 & -55 \\ 55 & -1 \end{pmatrix},$$

$$C_3 = \begin{pmatrix} -2 & 1 \\ 0 & -2 \end{pmatrix}.$$

In this figure, we can see the exponential decay to zero associated with the unequal real eigenvalues of C_1; the damped oscillation associated with the complex eigenvalues of C_2; and the initial growth of the time series due to the te^{-2t} term followed by exponential decay to zero in the equal eigenvalue C_3 example.

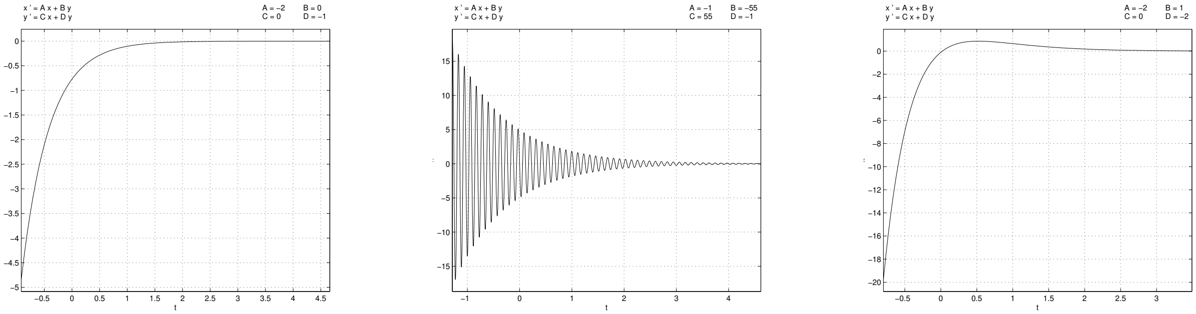

Figure 18: Time series for different sinks.

Sources Versus Sinks The explicit form of solutions to planar linear systems shows that solutions with initial conditions near the origin grow exponentially in forward time when the origin of (6.4.1) is a source. We can prove this point geometrically, as follows.

The phase planes of sources and sinks are almost the same; they have the same trajectories but the arrows are reversed. To verify this point, note that

$$\dot{X} = -CX \tag{6.4.2}$$

is a sink when (6.4.1) is a source; observe that the trajectories of solutions of (6.4.1) are the same as those of (6.4.2) — just with time running backwards. For let $X(t)$ be a solution to (6.4.1); then $X(-t)$ is a solution to (6.4.2). See Figure 19 for plots of $\dot{X} = BX$ and $\dot{X} = -BX$ where

$$B = \begin{pmatrix} -1 & -5 \\ 5 & -1 \end{pmatrix}. \tag{6.4.3}$$

So when we draw schematic phase portraits for sinks, we automatically know how to draw schematic phase portraits for sources. The trajectories are the same — but the arrows point in the opposite direction.

Phase Portraits for Saddles Next we discuss the phase portraits of linear saddles. Using pplane10, draw the phase portrait of the saddle

$$\begin{array}{rcl} \dot{x} & = & 2x + y \\ \dot{y} & = & -x - 3y, \end{array} \tag{6.4.4}$$

as in Figure 20. The important feature of saddles is that there are special trajectories (the eigendirections) that limit on the origin in either forward or backward time.

Definition 6.4.3. The *stable manifold* or *stable orbit* of a saddle consists of those trajectories that limit on the origin in forward time; the *unstable manifold* or *unstable orbit* of a saddle consists of those trajectories that limit on the origin in backward time.

Let $\lambda_1 < 0$ and $\lambda_2 > 0$ be the eigenvalues of a saddle with associated eigenvectors v_1 and v_2. The stable orbits are given by the solutions $X(t) = \pm e^{\lambda_1 t} v_1$ and the unstable orbits are given by the solutions $X(t) = \pm e^{\lambda_2 t} v_2$.

Stable and Unstable Orbits using pplane10 The program pplane10 is programmed to draw the stable and unstable orbits of a saddle on command. Although the principal use of this feature is seen when analyzing nonlinear systems, it is useful to introduce this feature here. As an

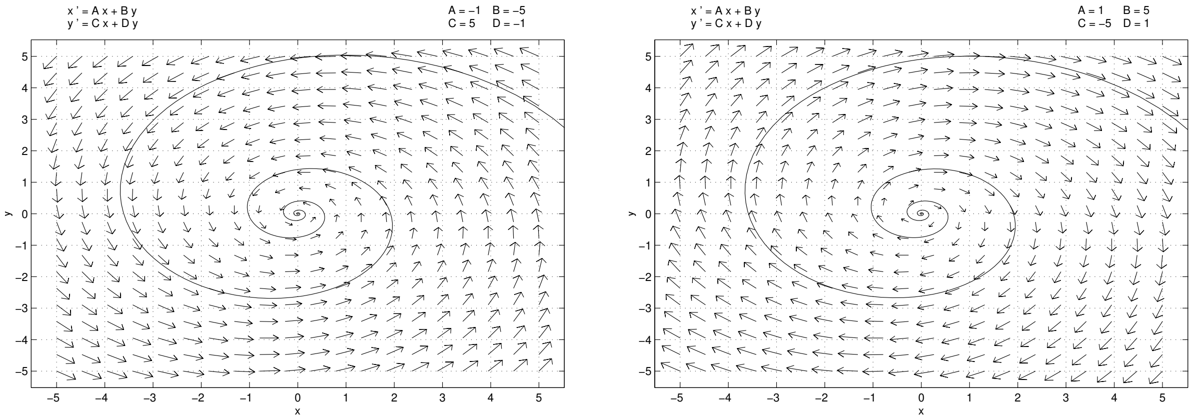

Figure 19: (Left) Sink $\dot{X} = BX$ where B is given in (6.4.3). (Right) Source $\dot{X} = -BX$.

example, load the linear system (6.4.4) into pplane10 and click on Proceed. Now pull down the PPLANE9 Options menu and click on Find an equilibrium. Click the cross hairs in the PPLANE9 Display window on a point near the origin; pplane10 responds by opening a new window — the PPLANE9 Equilibrium point data window — and by putting a small yellow circle about the origin. The circle indicates that the numerical algorithm programmed into pplane10 has detected an equilibrium near the chosen point. A new window opens and displays the message There is a saddle point at $(0, 0)$. This window also displays the coefficient matrix (called the *Jacobian* at the equilibrium) and its eigenvalues and eigenvectors. This process numerically verifies that the origin is a saddle (a fact that could have been verified in a more straightforward way).

Now pull down the PPLANE9 Options menu again and click on Plot stable and unstable orbits. Next click on the mouse when the cross hairs are within the yellow circle and pplane10 responds by drawing the stable and unstable orbits. The result is shown in Figure 20(left). On this figure we have also plotted one trajectory from each quadrant; thus obtaining the phase portrait of a saddle. On the right of Figure 20 we have plotted a time series of the first quadrant solution. Note how the x time series increases exponentially to $+\infty$ in forward time and the y time series decreases in forward time while going exponentially towards $-\infty$. The two time series together give the trajectory $(x(t), y(t))$ that in forward time is asymptotic to the line given by the unstable eigendirection.

Exercises

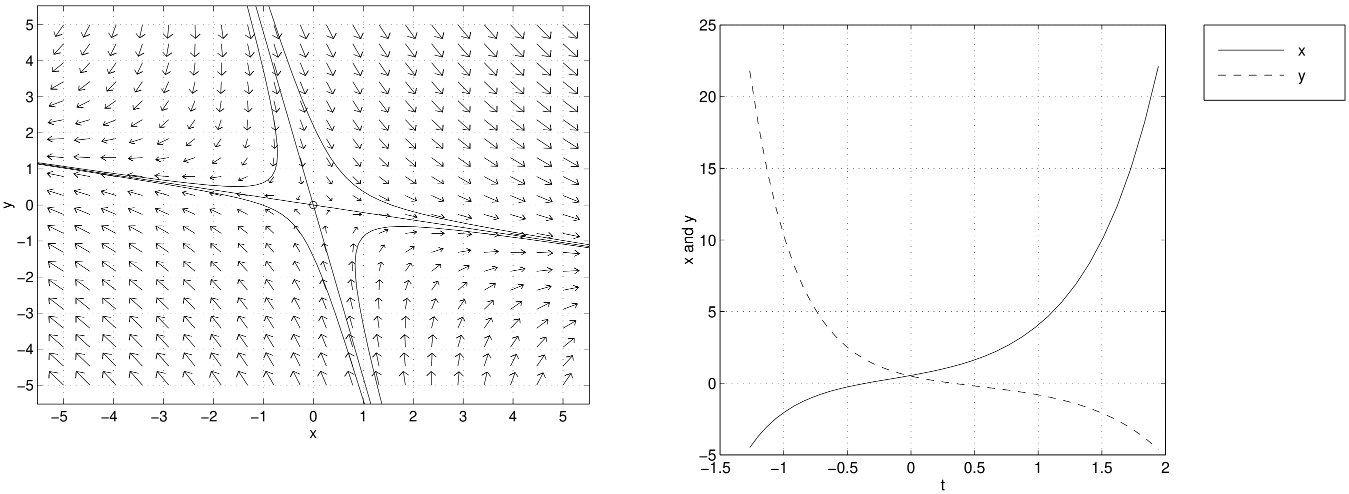

Figure 20: (Left) Saddle phase portrait. (Right) First quadrant solution time series.

6.5 *Matrix Exponentials

In Section 4.2 we showed that the solution of the single ordinary differential equation $\dot{x}(t) = \lambda x(t)$ with initial condition $x(0) = x_0$ is $x(t) = e^{t\lambda}x_0$ (see (4.1.4) in Chapter 4). In this section we show that we may write solutions of systems of equations in a similar form. In particular, we show that the solution to the linear system of ODEs

$$\frac{dX}{dt} = CX \tag{6.5.1}$$

with initial condition

$$X(0) = X_0,$$

where C is an $n \times n$ matrix and $X_0 \in \mathbb{R}^n$, is

$$X(t) = e^{tC}X_0. \tag{6.5.2}$$

In order to make sense of the solution (6.5.2) we need to understand matrix exponentials. More precisely, since tC is an $n \times n$ matrix for each $t \in \mathbb{R}$, we need to make sense of the expression e^L where L is an $n \times n$ matrix. For this we recall the form of the exponential function as a power series:

$$e^t = 1 + t + \frac{1}{2!}t^2 + \frac{1}{3!}t^3 + \frac{1}{4!}t^4 + \cdots .$$

In more compact notation we have

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

By analogy, define the *matrix exponential* e^L by

$$\begin{aligned} e^L &= I_n + L + \frac{1}{2!}L^2 + \frac{1}{3!}L^3 + \cdots \\ &= \sum_{k=0}^{\infty} \frac{1}{k!}L^k. \end{aligned} \tag{6.5.3}$$

In this formula $L^2 = LL$ is the matrix product of L with itself, and the power L^k is defined inductively by $L^k = LL^{k-1}$ for $k > 1$. Hence e^L is an $n \times n$ matrix and is the infinite sum of $n \times n$ matrices.

Remark: The infinite series for matrix exponentials (6.5.3) converges for all $n \times n$ matrices L. This fact is proved in Exercises **??** and **??**.

Using (6.5.3), we can write the matrix exponential of tC for each real number t. Since $(tC)^k = t^kC^k$ we obtain

$$\begin{aligned} e^{tC} &= I_n + tC + \frac{1}{2!}(tC)^2 + \frac{1}{3!}(tC)^3 + \cdots \\ &= I_n + tC + \frac{t^2}{2!}C^2 + \frac{t^3}{3!}C^3 + \cdots . \end{aligned} \tag{6.5.4}$$

Next we claim that

$$\frac{d}{dt}e^{tC} = Ce^{tC}. \tag{6.5.5}$$

We verify the claim by supposing that we can differentiate (6.5.4) term by term with respect to t. Then

$$\begin{aligned}\frac{d}{dt}e^{tC} &= \frac{d}{dt}(I_n) + \frac{d}{dt}(tC) + \frac{d}{dt}\left(\frac{t^2}{2!}C^2\right) + \frac{d}{dt}\left(\frac{t^3}{3!}C^3\right) + \\ &\quad \frac{d}{dt}\left(\frac{t^4}{4!}C^4\right) + \cdots \\ &= 0 + C + tC^2 + \frac{t^2}{2!}C^3 + \frac{t^3}{3!}C^4 + \cdots \\ &= C\left(I_n + tC + \frac{t^2}{2!}C^2 + \frac{t^3}{3!}C^3 + \cdots\right) \\ &= Ce^{tC}.\end{aligned}$$

It follows that the function $X(t) = e^{tC}X_0$ is a solution of (6.5.1) for each $X_0 \in \mathbb{R}^n$; that is,

$$\frac{d}{dt}X(t) = \frac{d}{dt}e^{tC}X_0 = Ce^{tC}X_0 = CX(t).$$

Since (6.5.3) implies that $e^{0C} = e^0 = I_n$, it follows that $X(t) = e^{tC}X_0$ is a solution of (6.5.1) with initial condition $X(0) = X_0$. This discussion shows that solving (6.5.1) in closed form is equivalent to finding a closed form expression for the matrix exponential e^{tC}.

Theorem 6.5.1. *The unique solution to the initial value problem*

$$\begin{aligned}\frac{dX}{dt} &= CX \\ X(0) &= X_0\end{aligned}$$

is

$$X(t) = e^{tC}X_0.$$

Proof Existence follows from the previous discussion. For uniqueness, suppose that $Y(t)$ is a solution to $\dot{Y} = CY$ with $Y(0) = X_0$. We claim that $Y(t) = X(t)$. Let $Z(t) = e^{-tC}Y(t)$ and use the product rule to compute

$$\frac{dZ}{dt} = -Ce^{-tC}Y(t) + e^{-tC}\frac{dY}{dt}(t) = e^{-tC}(-CY(t) + CY(t)) = 0$$

It follows that Z is constant in t and $Z(t) = Z(0) = Y(0) = X_0$ or $Y(t) = e^{tC}X_0 = X(t)$, as claimed. ∎

Similarity and Matrix Exponentials We introduce similarity at this juncture for the following reason: if C is a matrix that is similar to B, then e^C can be computed from e^B. More precisely:

Lemma 6.5.2. *Let C and B be $n \times n$ similar matrices, and let P be an invertible $n \times n$ matrix such that*

$$C = P^{-1}BP.$$

Then

$$e^C = P^{-1}e^BP. \tag{6.5.6}$$

Proof Note that for all powers of k we have

$$(P^{-1}BP)^k = P^{-1}B^kP.$$

Next verify (6.5.6) by computing

$$\begin{aligned}e^C &= \sum_{k=0}^{\infty}\frac{1}{k!}C^k = \sum_{k=0}^{\infty}\frac{1}{k!}(P^{-1}BP)^k \\ &= \sum_{k=0}^{\infty}\frac{1}{k!}P^{-1}B^kP = P^{-1}\left(\sum_{k=0}^{\infty}\frac{1}{k!}B^k\right)P = P^{-1}e^BP.\end{aligned}$$

∎

Explicit Computation of Matrix Exponentials We begin with the simplest computation of a matrix exponential.

(a) Let L be a multiple of the identity; that is, let $L = \alpha I_n$ where α is a real number. Then

$$e^{\alpha I_n} = e^{\alpha}I_n. \tag{6.5.7}$$

That is, $e^{\alpha I_n}$ is a scalar multiple of the identity. To verify (6.5.7), compute

$$\begin{aligned} e^{\alpha I_n} &= I_n + \alpha I_n + \frac{\alpha^2}{2!}I_n^2 + \frac{\alpha^3}{3!}I_n^3 + \cdots \\ &= (1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \cdots)I_n = e^{\alpha} I_n. \end{aligned}$$

(b) Let C be a 2×2 diagonal matrix,

$$C = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix},$$

where λ_1 and λ_2 are real constants. Then

$$e^{tC} = \begin{pmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{pmatrix}. \tag{6.5.8}$$

To verify (6.5.8) compute

$$\begin{aligned} e^{tC} &= I_2 + tC + \frac{t^2}{2!}C^2 + \frac{t^3}{3!}C^3 + \cdots \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} \lambda_1 t & 0 \\ 0 & \lambda_2 t \end{pmatrix} + \begin{pmatrix} \frac{t^2}{2!}\lambda_1^2 & 0 \\ 0 & \frac{t^2}{2!}\lambda_2^2 \end{pmatrix} + \cdots \\ &= \begin{pmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{pmatrix}. \end{aligned}$$

(c) Suppose that

$$C = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Then

$$e^{tC} = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}. \tag{6.5.9}$$

We begin this computation by observing that

$$C^2 = -I_2, \quad C^3 = -C, \quad \text{and} \quad C^4 = I_n.$$

Therefore, by collecting terms of odd and even power in the series expansion for the matrix exponential we obtain

$$\begin{aligned} e^{tC} &= I_2 + tC + \frac{t^2}{2!}C^2 + \frac{t^3}{3!}C^3 + \cdots \\ &= I_2 + tC - \frac{t^2}{2!}I_2 - \frac{t^3}{3!}C + \cdots \\ &= \left(1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \frac{t^6}{6!} + \cdots\right) I_2 + \\ &\quad \left(t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \cdots\right) C \\ &= (\cos t)I_2 + (\sin t)C \\ &= \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}. \end{aligned}$$

In this computation we have used the fact that the trigonometric functions $\cos t$ and $\sin t$ have the power series expansions:

$$\begin{aligned} \cos t &= 1 - \frac{1}{2!}t^2 + \frac{1}{4!}t^4 + \cdots = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!}t^{2k}, \\ \sin t &= t - \frac{1}{3!}t^3 + \frac{1}{5!}t^5 + \cdots = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!}t^{2k+1}. \end{aligned}$$

See Exercise **??** for an alternative proof of (6.5.9).

To compute the matrix exponential MATLAB provides the command `expm`. We use this command to compute the matrix exponential e^{tC} for

$$C = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad t = \frac{\pi}{4}.$$

Type

```
C = [0, -1; 1, 0];
t = pi/4;
expm(t*C)
```

that gives the answer

```
ans =
    0.7071   -0.7071
    0.7071    0.7071
```

Indeed, this is precisely what we expect by (6.5.9), since

$$\cos\left(\frac{\pi}{4}\right) = \sin\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} \approx 0.70710678.$$

(d) Let

$$C = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

Then

$$e^{tC} = I_2 + tC = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}, \tag{6.5.10}$$

since $C^2 = 0$.

Exercises

6.6 *The Cayley Hamilton Theorem

The Jordan normal form theorem (Theorem 6.3.4) for real 2×2 matrices states that every 2×2 matrix is similar to one of the matrices in Table 2. We use this theorem to prove the Cayley Hamilton theorem for 2×2 matrices and then use the Cayley Hamilton theorem to present another method for computing solutions to planar linear systems of differential equations in the case of real equal eigenvalues.

The Cayley Hamilton theorem states that a matrix satisfies its own characteristic polynomial. More precisely:

Theorem 6.6.1 (Cayley Hamilton Theorem)**.** *Let A be a 2×2 matrix and let*

$$p_A(\lambda) = \lambda^2 + a\lambda + b$$

be the characteristic polynomial of A. Then

$$p_A(A) = A^2 + aA + bI_2 = 0.$$

Proof Suppose $B = P^{-1}AP$ and A are similar matrices. We claim that if $p_A(A) = 0$, then $p_B(B) = 0$. To verify this claim, recall from Lemma 6.3.3 that $p_A = p_B$ and calculate

$$\begin{aligned} p_B(B) &= p_A(P^{-1}AP) = (P^{-1}AP)^2 + aP^{-1}AP + bI_2 \\ &= P^{-1}p_A(A)P = 0. \end{aligned}$$

Theorem 6.3.4 classifies 2×2 matrices up to similarity. Thus, we need only verify this theorem for the matrices

$$C = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, D = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix}, E = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix},$$

that is, we need to verify that

$$p_C(C) = 0 \qquad p_D(D) = 0 \qquad p_E(E) = 0.$$

Using the fact that $p_A(\lambda) = \lambda^2 - \mathrm{tr}(A)\lambda + \det(A)$, we see that

$$\begin{aligned} p_C(\lambda) &= (\lambda - \lambda_1)(\lambda - \lambda_2) \\ p_D(\lambda) &= \lambda^2 - 2\sigma\lambda + (\sigma^2 + \tau^2) \\ p_E(\lambda) &= (\lambda - \lambda_1)^2. \end{aligned}$$

It now follows that

$$\begin{aligned} p_C(C) &= (C - \lambda_1 I_2)(C - \lambda_2 I_2) \\ &= \begin{pmatrix} 0 & 0 \\ 0 & \lambda_2 - \lambda_1 \end{pmatrix}\begin{pmatrix} \lambda_1 - \lambda_2 & 0 \\ 0 & 0 \end{pmatrix} \\ &= 0, \end{aligned}$$

and

$$\begin{aligned} p_D(D) = \begin{pmatrix} \sigma^2 - \tau^2 & -2\sigma\tau \\ 2\sigma\tau & \sigma^2 - \tau^2 \end{pmatrix} - \\ 2\sigma\begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix} + \\ (\sigma^2 + \tau^2)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = 0, \end{aligned}$$

and

$$p_E(E) = (E - \lambda_1 I_2)^2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^2 = 0.$$

■

The Example with Equal Eigenvalues Revisited When the eigenvalues $\lambda_1 = \lambda_2$, the closed form solution of $\dot{X} = CX$ is a straightforward formula

$$X(t) = e^{\lambda_1 t}(I_2 + tN) \tag{6.6.1}$$

where $N = C - \lambda_1 I_2$.

Note that when using (6.6.1), it is not necessary to compute the eigenvector or generalized eigenvector of C, and this is a substantial simplification.

Verification of (6.6.1) We use the Cayley-Hamilton theorem to verify (6.6.1) as follows. Specifically, since C is assumed to have a double eigenvalue λ_1, it follows that

$$N = C - \lambda_1 I_2$$

has zero as a double eigenvalue. Hence, the characteristic polynomial $p_N(\lambda) = \lambda^2$ and the Cayley Hamilton theorem implies that $N^2 = 0$. Therefore,

$$CX(t) = e^{\lambda_1 t} C(I_2 + tN)X_0 = e^{\lambda_1 t}(\lambda_1 I_2 + N)(I_2 + tN)X_0$$

Thus, using $N^2 = 0$, we see that

$$CX(t) = e^{\lambda_1 t}(\lambda_1 I_2 + t\lambda_1 N + N)X_0 = \dot{X}(t),$$

as desired

Let us reconsider the system of differential equations (6.2.17)

$$\frac{dX}{dt} = \begin{pmatrix} 1 & -1 \\ 9 & -5 \end{pmatrix} X = CX$$

with initial value

$$X_0 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

The eigenvalues of C are real and equal to $\lambda_1 = -2$.

We may write

$$C = \lambda_1 I_2 + N = -2I_2 + N,$$

where

$$N = \begin{pmatrix} 3 & -1 \\ 9 & -3 \end{pmatrix}.$$

It follows from (6.6.1) that

$$e^{tC} = e^{-2t}\left(I_2 + t\begin{pmatrix} 3 & -1 \\ 9 & -3 \end{pmatrix}\right) = e^{-2t}\begin{pmatrix} 1+3t & -t \\ 9t & 1-3t \end{pmatrix}. \tag{6.6.2}$$

Hence the solution to the initial value problem is:

$$\begin{aligned} X(t) = e^{tC}X_0 &= e^{-2t}\begin{pmatrix} 1+3t & -t \\ 9t & 1-3t \end{pmatrix}\begin{pmatrix} 2 \\ 3 \end{pmatrix} \\ &= e^{-2t}\begin{pmatrix} 2+3t \\ 3+9t \end{pmatrix}. \end{aligned}$$

Exercises

6.7 *Second Order Equations

A second order constant coefficient homogeneous differential equation is a differential equation of the form:

$$\ddot{x} + b\dot{x} + ax = 0, \tag{6.7.1}$$

where a and b are real numbers.

Newton's Second Law *Newton's second law of motion* is a second order ordinary differential equation, and for this reason second order equations arise naturally in mechanical systems. Newton's second law states that

$$F = ma \tag{6.7.2}$$

where F is force, m is mass, and a is acceleration.

Newton's Second Law and Particle Motion on a Line For a point mass moving along a line, (6.7.2) is

$$F = m\frac{d^2x}{dt^2}, \tag{6.7.3}$$

where $x(t)$ is the position of the point mass at time t. For example, suppose that a particle of mass m is falling towards the earth. If we let g be the gravitational constant and if we ignore all forces except gravitation, then the force acting on that particle is $F = -mg$. In this case Newton's second law leads to the second order ordinary differential equation

$$\frac{d^2x}{dt^2} + g = 0. \tag{6.7.4}$$

Newton's Second Law and the Motion of a Spring As a second example, consider the spring model pictured in Figure 21. Assume that the spring has zero mass and that an object of mass m is attached to the end of the spring. Let L be the natural length of the spring, and let $x(t)$ measure the distance that the spring is extended (or compressed). It follows from Newton's Law that (6.7.3) is satisfied. Hooke's law states that the force F acting on a spring is

$$F = -\kappa x,$$

where κ is a positive constant. If the spring is damped by sliding friction, then

$$F = -\kappa x - \mu\frac{dx}{dt},$$

where μ is also a positive constant. Suppose, in addition, that an external force $F_{ext}(t)$ also acts on the mass and that that force is time-dependent. Then the entire force acting on the mass is

$$F = -\kappa x - \mu\frac{dx}{dt} + F_{ext}(t).$$

By Newton's second law, the motion of the mass is described by

$$m\frac{d^2x}{dt^2} + \mu\frac{dx}{dt} + \kappa x = F_{ext}(t), \tag{6.7.5}$$

which is again a second order ordinary differential equation.

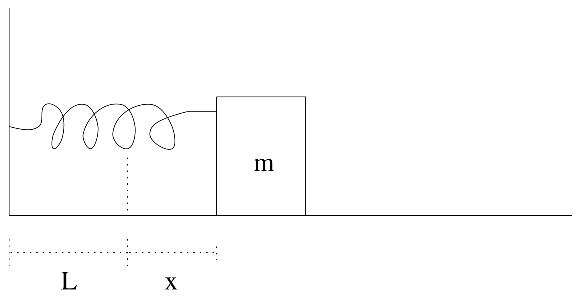

Figure 21: Hooke's Law spring.

A Reduction to a First Order System There is a simple trick that reduces a single linear second order differential equation to a system of two linear first order equations. For example, consider the linear homogeneous ordinary differential equation (6.7.1). To reduce this second order equation to a first order system, just set $y = \dot{x}$. Then (6.7.1) becomes

$$\dot{y} + by + ax = 0.$$

It follows that if $x(t)$ is a solution to (6.7.1) and $y(t) = \dot{x}(t)$, then $(x(t), y(t))$ is a solution to

$$\begin{array}{rcl} \dot{x} & = & y \\ \dot{y} & = & -ax - by. \end{array} \tag{6.7.6}$$

We can rewrite (6.7.6) as

$$\dot{X} = QX.$$

where

$$Q = \begin{pmatrix} 0 & 1 \\ -a & -b \end{pmatrix}. \tag{6.7.7}$$

Note that if $(x(t), y(t))$ is a solution to (6.7.6), then $x(t)$ is a solution to (6.7.1). Thus solving the single second order linear equation is exactly the same as solving the corresponding first order linear system.

The Initial Value Problem To solve the homogeneous system (6.7.6) we need to specify two initial conditions $X(0) = (x(0), y(0))^t$. It follows that to solve the single second order equation we need to specify two initial conditions $x(0)$ and $\dot{x}(0)$; that is, we need to specify both initial position and initial velocity.

The General Solution There are two ways in which we can solve the second order homogeneous equation (6.7.1). First, we know how to solve the system (6.7.6) by finding the eigenvalues and eigenvectors of the coefficient matrix Q in (6.7.7). Second, we know from the general theory of planar systems that solutions will have the form $x(t) = e^{\lambda_0 t}$ for some scalar λ_0. We need only determine the values of λ_0 for which we get solutions to (6.7.1).

We now discuss the second approach. Suppose that $x(t) = e^{\lambda_0 t}$ is a solution to (6.7.1). Substituting this form of $x(t)$ in (6.7.1) yields the equation

$$\left(\lambda_0^2 + b\lambda_0 + a\right) e^{\lambda_0 t} = 0.$$

So $x(t) = e^{\lambda_0 t}$ is a solution to (6.7.1) precisely when $p_Q(\lambda_0) = 0$, where

$$p_Q(\lambda) = \lambda^2 + b\lambda + a \tag{6.7.8}$$

is the characteristic polynomial of the matrix Q in (6.7.7).

Suppose that λ_1 and λ_2 are distinct real roots of p_Q. Then the general solution to (6.7.1) is

$$x(t) = \alpha_1 e^{\lambda_1 t} + \alpha_2 e^{\lambda_2 t},$$

where $\alpha_j \in \mathbb{R}$.

An Example with Distinct Real Eigenvalues For example, solve the initial value problem

$$\ddot{x} + 3\dot{x} + 2x = 0 \tag{6.7.9}$$

with initial conditions $x(0) = 0$ and $\dot{x}(0) = -2$. The characteristic polynomial is

$$p_Q(\lambda) = \lambda^2 + 3\lambda + 2 = (\lambda + 2)(\lambda + 1),$$

whose roots are $\lambda_1 = -1$ and $\lambda_2 = -2$. So the general solution to (6.7.9) is

$$x(t) = \alpha_1 e^{-t} + \alpha_2 e^{-2t}$$

To find the precise solution we need to solve

$$\begin{array}{rclcl} x(0) & = & \alpha_1 + \alpha_2 & = & 0 \\ \dot{x}(0) & = & -\alpha_1 - 2\alpha_2 & = & -2 \end{array}$$

So $\alpha_1 = -2$, $\alpha_2 = 2$, and the solution to the initial value problem for (6.7.9) is

$$x(t) = -2e^{-t} + 2e^{-2t}$$

An Example with Complex Conjugate Eigenvalues Consider the differential equation

$$\ddot{x} - 2\dot{x} + 5x = 0. \tag{6.7.10}$$

The roots of the characteristic polynomial associated to (6.7.10) are $\lambda_1 = 1 + 2i$ and $\lambda_2 = 1 - 2i$. It follows from the discussion in the previous section that the general solution to (6.7.10) is

$$x(t) = \mathrm{Re}\left(\alpha_1 e^{\lambda_1 t} + \alpha_2 e^{\lambda_2 t}\right)$$

where α_1 and α_2 are complex scalars. Indeed, we can rewrite this solution in real form (using Euler's formula) as

$$x(t) = e^t\left(\beta_1 \cos(2t) + \beta_2 \sin(2t)\right),$$

for real scalars β_1 and β_2.

In general, if the roots of the characteristic polynomial are $\sigma \pm i\tau$, then the general solution to the differential equation is:

$$x(t) = e^{\sigma t}\left(\beta_1 \cos(\tau t) + \beta_2 \sin(\tau t)\right).$$

An Example with Multiple Eigenvalues Note that the coefficient matrix Q of the associated first order system in (6.7.7) is never a multiple of I_2. It follows from the previous section that when the roots of the characteristic polynomial are real and equal, the general solution has the form

$$x(t) = \alpha_1 e^{\lambda_1 t} + \alpha_2 t e^{\lambda_2 t}.$$

Summary It follows from this discussion that solutions to second order homogeneous linear equations are either a linear combination of two exponentials (real unequal eigenvalues), $\alpha + \beta t$ times one exponential (real equal eigenvalues), or a time periodic function times an exponential (complex eigenvalues).

In particular, if the real part of the complex eigenvalues is zero, then the solution is time periodic. The frequency of this periodic solution is often called the *internal frequency*, a point that is made more clearly in the next example.

Solving the Spring Equation Consider the equation for the frictionless spring without external forcing. From (6.7.5) we get

$$m\ddot{x} + \kappa x = 0. \tag{6.7.11}$$

where $\kappa > 0$. The roots are $\lambda_1 = \sqrt{\frac{\kappa}{m}}i$ and $\lambda_2 = -\sqrt{\frac{\kappa}{m}}i$. So the general solution is

$$x(t) = \alpha \cos(\tau t) + \beta \sin(\tau t),$$

where $\tau = \sqrt{\frac{\kappa}{m}}$. Under these assumptions the motion of the spring is time periodic with period $\frac{2\pi}{\tau}$ or internal frequency $\frac{\tau}{2\pi}$. In particular, the solution satisfying initial conditions $x(0) = 1$ and $\dot{x}(0) = 0$ (the spring is extended one unit in distance and released with no initial velocity) is

$$x(t) = \cos(\tau t).$$

The graph of this function when $\tau = 1$ is given on the left in Figure 22.

If a small amount of friction is added, then the spring equation is

$$m\ddot{x} + \mu\dot{x} + \kappa x = 0$$

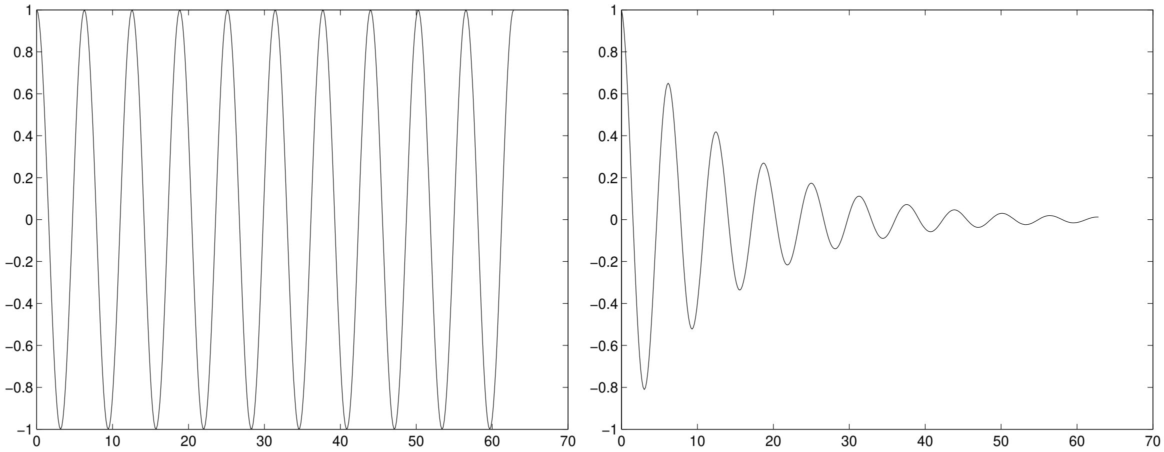

Figure 22: (Left) Graph of solution to undamped spring equation with initial conditions $x(0) = 1$ and $\dot{x}(0) = 0$. (Right) Graph of solution to damped spring equation with the same initial conditions.

where $\mu > 0$ is small. Since the eigenvalues of the characteristic polynomial are $\lambda = \sigma \pm i\tau$ where

$$\sigma = -\frac{\mu}{2m} < 0 \quad \text{and} \quad \tau = \sqrt{\frac{\kappa}{m} - \left(\frac{\mu}{2m}\right)^2},$$

the general solution is

$$x(t) = e^{\sigma t}(\alpha\cos(\tau t) + \beta\sin(\tau t)).$$

Since $\sigma < 0$, these solutions oscillate but damp down to zero. In particular, the solution satisfying initial conditions $x(0) = 1$ and $\dot{x}(0) = 0$ is

$$x(t) = e^{-\mu t/2m}\left(\cos(\tau t) - \frac{\mu}{2m\tau}\sin(\tau t)\right).$$

The graph of this solution when $\tau = 1$ and $\dfrac{\mu}{2m} = 0.07$ is given in Figure 22 (right). Compare the solutions for the undamped and damped springs.

Exercises

7 Determinants and Eigenvalues

In Section 3.8 we introduced determinants for 2×2 matrices A. There we showed that the determinant of A is nonzero if and only if A is invertible. In Section 4.6 we saw that the eigenvalues of A are the roots of its characteristic polynomial, and that its characteristic polynomial is just the determinant of a related matrix, namely, $p_A(\lambda) = \det(A - \lambda I_2)$.

In Section 7.1 we generalize the concept of determinants to $n \times n$ matrices, and in Section 7.2 we use determinants to show that every $n \times n$ matrix has exactly n eigenvalues — the roots of its characteristic polynomial. Properties of eigenvalues are also discussed in detail in Section 7.2. Certain details concerning determinants are deferred to Appendix 7.4.

7.1 Determinants

There are several equivalent ways to introduce determinants — none of which are easily motivated. We prefer to define determinants through the properties they satisfy rather than by formula. These properties actually enable us to compute determinants of $n\times n$ matrices where $n>3$, which further justifies the approach. Later on, we will give an inductive formula (7.1.9) for computing the determinant.

Definition 7.1.1. A *determinant function* of a square $n\times n$ matrix A is a real number $D(A)$ that satisfies three properties:

(a) If $A=(a_{ij})$ is lower triangular, then $D(A)$ is the product of the diagonal entries; that is,

$$D(A)=a_{11}\cdot\cdots\cdot a_{nn}.$$

(b) $D(A^t)=D(A)$.

(c) Let B be an $n\times n$ matrix. Then

$$D(AB)=D(A)D(B). \tag{7.1.1}$$

Theorem 7.1.2. *There exists a unique determinant function* det *satisfying the three properties of Definition 7.1.1.*

We will show that it is possible to compute the determinant of any $n\times n$ matrix using Definition 7.1.1. Here we present a few examples:

Lemma 7.1.3. *Let A be an $n\times n$ matrix.*

(a) Let $c\in\mathbb{R}$ be a scalar. Then $D(cA)=c^nD(A)$.

(b) If all of the entries in either a row or a column of A are zero, then $D(A)=0$.

Proof (a) Note that Definition 7.1.1(a) implies that $D(cI_n)=c^n$. It follows from (7.1.1) that

$$D(cA)=D(cI_nA)=D(cI_n)D(A)=c^nD(A).$$

(b) Definition 7.1.1(b) implies that it suffices to prove this assertion when one row of A is zero. Suppose that the i^{th} row of A is zero. Let J be an $n\times n$ diagonal matrix with a 1 in every diagonal entry except the i^{th} diagonal entry which is 0. A matrix calculation shows that $JA=A$. It follows from Definition 7.1.1(a) that $D(J)=0$ and from (7.1.1) that $D(A)=0$. ■

Determinants of 2×2 Matrices Before discussing how to compute determinants, we discuss the special case of 2×2 matrices. Recall from (3.8.2) of Section 3.8 that when

$$A=\begin{pmatrix} a & b\\ c & d\end{pmatrix}$$

we defined

$$\det(A)=ad-bc. \tag{7.1.2}$$

We check that (7.1.2) satisfies the three properties in Definition 7.1.1. Observe that when A is lower triangular, then $b=0$ and $\det(A)=ad$. So (a) is satisfied. It is straightforward to verify (b). We already verified (c) in Chapter 3, Proposition 3.8.2.

It is less obvious perhaps — but true nonetheless — that the three properties of $D(A)$ actually force the determinant of 2×2 matrices to be given by formula (7.1.2). We begin by showing that Definition 7.1.1 implies that

$$D\begin{pmatrix} 0 & 1\\ 1 & 0\end{pmatrix}=-1. \tag{7.1.3}$$

We verify (7.1.3) by observing that

$$\begin{pmatrix} 0 & 1\\ 1 & 0\end{pmatrix}$$

equals

$$\begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \quad (7.1.4)$$

Hence property (c), (a) and (b) imply that

$$D\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = 1 \cdot 1 \cdot (-1) \cdot 1 = -1.$$

It is helpful to interpret the matrices in (7.1.4) as elementary row operations. Then (7.1.4) states that swapping two rows in a 2×2 matrix is the same as performing the following row operations in order:

- add the 2^{nd} row to the 1^{st} row;
- multiply the 2^{nd} row by -1;
- add the 1^{st} row to the 2^{nd} row; and
- subtract the 2^{nd} row from the 1^{st} row.

Suppose that $d \neq 0$. Then

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & \frac{b}{d} \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \frac{ad-bc}{d} & 0 \\ c & d \end{pmatrix}.$$

It follows from properties (c), (b) and (a) that

$$D(A) = \frac{ad-bc}{d}d = ad - bc = \det(A),$$

as claimed.

Now suppose that $d = 0$ and note that

$$A = \begin{pmatrix} a & b \\ c & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} c & 0 \\ a & b \end{pmatrix}.$$

Using (7.1.3) we see that

$$D(A) = -D\begin{pmatrix} c & 0 \\ a & b \end{pmatrix} = -bc = \det(A),$$

as desired.

We have verified that the only possible determinant function for 2×2 matrices is the determinant function defined by (7.1.2).

Row Operations are Invertible Matrices

Proposition 7.1.4. *Let A and B be $m \times n$ matrices where B is obtained from A by a single elementary row operation. Then there exists an invertible $m \times m$ matrix R such that $B = RA$.*

Proof First consider multiplying the j^{th} row of A by the nonzero constant c. Let R be the diagonal matrix whose j^{th} entry on the diagonal is c and whose other diagonal entries are 1. Then the matrix RA is just the matrix obtained from A by multiplying the j^{th} row of A by c. Note that R is invertible when $c \neq 0$ and that R^{-1} is the diagonal matrix whose j^{th} entry is $\frac{1}{c}$ and whose other diagonal entries are 1. For example

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 2a_{31} & 2a_{32} & 2a_{33} \end{pmatrix},$$

multiplies the 3^{rd} row by 2.

Next we show that the elementary row operation that swaps two rows may also be thought of as matrix multiplication. Let $R = (r_{kl})$ be the matrix that deviates from the identity matrix by changing in the four entries:

$$\begin{aligned} r_{ii} &= 0 \\ r_{jj} &= 0 \\ r_{ij} &= 1 \\ r_{ji} &= 1 \end{aligned}$$

A calculation shows that RA is the matrix obtained from

A by swapping the i^{th} and j^{th} rows. For example,

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \\ a_{11} & a_{12} & a_{13} \end{pmatrix},$$

which swaps the 1^{st} and 3^{rd} rows. Another calculation shows that $R^2 = I_n$ and hence that R is invertible since $R^{-1} = R$.

Finally, we claim that adding c times the i^{th} row of A to the j^{th} row of A can be viewed as matrix multiplication. Let $E_{k\ell}$ be the matrix all of whose entries are 0 except for the entry in the k^{th} row and ℓ^{th} column which is 1. Then $R = I_n + cE_{ij}$ has the property that RA is the matrix obtained by adding c times the j^{th} row of A to the i^{th} row. We can verify by multiplication that R is invertible and that $R^{-1} = I_n - cE_{ij}$. More precisely,

$$(I_n + cE_{ij})(I_n - cE_{ij}) = I_n + cE_{ij} - cE_{ij} - c^2E_{ij}^2 = I_n,$$

since $E_{ij}^2 = O$ for $i \neq j$. For example,

$$(I_3 + 5E_{12})A = \begin{pmatrix} 1 & 5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} + 5a_{21} & a_{12} + 5a_{22} & a_{13} + 5a_{23} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix},$$

adds 5 times the 2^{nd} row to the 1^{st} row. ■

Determinants of Elementary Row Matrices

Lemma 7.1.5. *(a) The determinant of the matrix that adds a multiple of one row to another is* 1.

(b) The determinant of the matrix that multiplies one row by c is c.

(c) The determinant of a swap matrix is -1.

Proof (a) The matrix that adds a multiple of one row to another is triangular (either upper or lower) and has 1's on the diagonal. Thus property (a) in Definition 7.1.1 implies that the determinants of these matrices are equal to 1.

(b) The matrix that multiplies the i^{th} row by $c \neq 0$ is a diagonal matrix all of whose diagonal entries are 1 except for $a_{ii} = c$. Again property (a) implies that the determinant of this matrix is $c \neq 0$.

(c) The matrix that swaps the i^{th} row with the j^{th} row is the product of four matrices of types (a) and (b). To see this let A be an $n \times n$ matrix whose i^{th} row vector is a_i. Then perform the following four operations in order:

Operation	Result		Matrix
Add r_i to r_j	$r_i = a_i$	$r_j = a_i + a_j$	B_1
Multiply r_i by -1	$r_j = -a_i$	$r_j = a_i + a_j$	B_2
Add r_j to r_i	$r_i = a_j$	$r_j = a_i + a_j$	B_3
Subtract r_i from r_j	$r_i = a_j$	$r_j = a_i$	B_4

It follows that the swap matrix equals $B_4B_3B_2B_1$. Therefore

$$\begin{aligned}\det(\text{swap}) &= \det(B_4)\det(B_3)\det(B_2)\det(B_1) \\ &= (1)(-1)(1)(1) = -1.\end{aligned}$$

■

Computation of Determinants We now show how to compute the determinant of any $n \times n$ matrix A using elementary row operations and Definition 7.1.1. It follows from Proposition 7.1.4 that every elementary row operation on A may be performed by premultiplying A by an elementary row matrix.

For each matrix A there is a unique reduced echelon form matrix E and a sequence of elementary row matrices

$R_1 \dots R_s$ such that

$$E = R_s \cdots R_1 A. \tag{7.1.5}$$

It follows from Definition 7.1.1(c) that we can compute the determinant of A once we know the determinants of reduced echelon form matrices and the determinants of elementary row matrices. In particular

$$D(A) = \frac{D(E)}{D(R_1) \cdots D(R_s)}. \tag{7.1.6}$$

It is easy to compute the determinant of any matrix in reduced echelon form using Definition 7.1.1(a) since all reduced echelon form $n \times n$ matrices are upper triangular. Lemma 7.1.5 tells us how to compute the determinants of elementary row matrices. This discussion proves:

Proposition 7.1.6. *If a determinant function exists for $n \times n$ matrices, then it is unique. We call the unique determinant function* det.

We still need to show that determinant functions exist when $n > 2$. More precisely, we know that the reduced echelon form matrix E is uniquely defined from A (Chapter 2, Theorem 2.4.9), but there is more than one way to perform elementary row operations on A to get to E. Thus, we can write A in the form (7.1.6) in many different ways, and these different decompositions might lead to different values for $\det A$. (They don't.)

An Example of Determinants by Row Reduction As a practical matter we row reduce a square matrix A by premultiplying A by an elementary row matrix R_j. Thus

$$\det(A) = \frac{1}{\det(R_j)} \det(R_j A). \tag{7.1.7}$$

We use this approach to compute the determinant of the 4×4 matrix

$$A = \begin{pmatrix} 0 & 2 & 10 & -2 \\ 1 & 2 & 4 & 0 \\ 1 & 6 & 1 & -2 \\ 2 & 1 & 1 & 0 \end{pmatrix}.$$

The idea is to use (7.1.7) to keep track of the determinant while row reducing A to upper triangular form. For instance, swapping rows changes the sign of the determinant; so

$$\det(A) = -\det \begin{pmatrix} 1 & 2 & 4 & 0 \\ 0 & 2 & 10 & -2 \\ 1 & 6 & 1 & -2 \\ 2 & 1 & 1 & 0 \end{pmatrix}.$$

Adding multiples of one row to another leaves the determinant unchanged; so

$$\det(A) = -\det \begin{pmatrix} 1 & 2 & 4 & 0 \\ 0 & 2 & 10 & -2 \\ 0 & 4 & -3 & -2 \\ 0 & -3 & -7 & 0 \end{pmatrix}.$$

Multiplying a row by a scalar c corresponds to an elementary row matrix whose determinant is c. To make sure that we do not change the value of $\det(A)$, we have to divide the determinant by c as we multiply a row of A by c. So as we divide the second row of the matrix by 2, we multiply the whole result by 2, obtaining

$$\det(A) = -2 \det \begin{pmatrix} 1 & 2 & 4 & 0 \\ 0 & 1 & 5 & -1 \\ 0 & 4 & -3 & -2 \\ 0 & -3 & -7 & 0 \end{pmatrix}.$$

We continue row reduction by zeroing out the last two entries in the 2^{nd} column, obtaining

$$\begin{aligned}\det(A) &= -2\det\begin{pmatrix} 1 & 2 & 4 & 0 \\ 0 & 1 & 5 & -1 \\ 0 & 0 & -23 & 2 \\ 0 & 0 & 8 & -3 \end{pmatrix} \\ &= 46\det\begin{pmatrix} 1 & 2 & 4 & 0 \\ 0 & 1 & 5 & -1 \\ 0 & 0 & 1 & -\frac{2}{23} \\ 0 & 0 & 8 & -3 \end{pmatrix}.\end{aligned}$$

Thus

$$\det(A) = 46\det\begin{pmatrix} 1 & 2 & 4 & 0 \\ 0 & 1 & 5 & -1 \\ 0 & 0 & 1 & -\frac{2}{23} \\ 0 & 0 & 0 & -\frac{53}{23} \end{pmatrix} = -106.$$

Determinants and Inverses We end this subsection with an important observation about the determinant function. This observation generalizes to dimension n Corollary 3.8.3 of Chapter 3.

Theorem 7.1.7. *An $n\times n$ matrix A is invertible if and only if $\det(A)\neq 0$. Moreover, if A^{-1} exists, then*

$$\det A^{-1} = \frac{1}{\det A}. \tag{7.1.8}$$

Proof If A is invertible, then

$$\det(A)\det(A^{-1}) = \det(AA^{-1}) = \det(I_n) = 1.$$

Thus $\det(A)\neq 0$ and (7.1.8) is valid. In particular, the determinants of elementary row matrices are nonzero, since they are all invertible. (This point was proved by direct calculation in Lemma 7.1.5.)

If A is singular, then A is row equivalent to a non-identity reduced echelon form matrix E whose determinant is zero (since E is upper triangular and its last diagonal entry is zero). So it follows from (7.1.5) that

$$0 = \det(E) = \det(R_1)\cdots\det(R_s)\det(A)$$

Since $\det(R_j)\neq 0$, it follows that $\det(A) = 0$. ■

Corollary 7.1.8. *If the rows of an $n\times n$ matrix A are linearly dependent (for example, if one row of A is a scalar multiple of another row of A), then $\det(A) = 0$.*

An Inductive Formula for Determinants In this subsection we present an inductive formula for the determinant — that is, we assume that the determinant is known for square $(n-1)\times(n-1)$ matrices and use this formula to define the determinant for $n\times n$ matrices. This inductive formula is called *expansion by cofactors*.

Let $A = (a_{ij})$ be an $n\times n$ matrix. Let A_{ij} be the $(n-1)\times(n-1)$ matrix formed from A by deleting the i^{th} row and the j^{th} column. The matrices A_{ij} are called *cofactor* matrices of A.

Inductively we define the determinant of an $n\times n$ matrix A by:

$$\begin{aligned}\det(A) &= \sum_{j=1}^{n}(-1)^{1+j}a_{1j}\det(A_{1j}) \\ &= a_{11}\det(A_{11}) - a_{12}\det(A_{12}) + \cdots \\ &\quad + (-1)^{n+1}a_{1n}\det(A_{1n}).\end{aligned} \tag{7.1.9}$$

In Appendix 7.4 we show that the determinant function defined by (7.1.9) satisfies all properties of a determinant function. Formula (7.1.9) is also called *expansion by cofactors along the 1^{st} row*, since the a_{1j} are taken from the 1^{st} row of A. Since $\det(A) = \det(A^t)$, it follows that if (7.1.9) is valid as an inductive definition of determinant, then expansion by cofactors along the 1^{st} column is also valid. That is,

$$\det(A) = a_{11}\det(A_{11}) - a_{21}\det(A_{21}) + \cdots + (-1)^{n+1}a_{n1}\det(A_{n1}) \quad (7.1.10)$$

We now explore some of the consequences of this definition, beginning with determinants of small matrices. For example, Definition 7.1.1(a) implies that the determinant of a 1×1 matrix is just

$$\det(a) = a.$$

Therefore, using (7.1.9), the determinant of a 2×2 matrix is:

$$\det\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = a_{11}\det(a_{22}) - a_{12}\det(a_{21}) = a_{11}a_{22} - a_{12}a_{21},$$

which is just the formula for determinants of 2×2 matrices given in (7.1.2).

Similarly, we can now find a formula for the determinant of 3×3 matrices A as follows.

$$\begin{aligned} \det(A) = a_{11}\det\begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} - a_{12}\det\begin{pmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{pmatrix} \\ + a_{13}\det\begin{pmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \\ = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} \\ - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{31}. \quad (7.1.11) \end{aligned}$$

As an example, compute

$$\det\begin{pmatrix} 2 & 1 & 4 \\ 1 & -1 & 3 \\ 5 & 6 & -2 \end{pmatrix}$$

using formula (7.1.11) as

$$\begin{aligned} 2(-1)(-2) + 1\cdot3\cdot5 + 4\cdot6\cdot1 - 4(-1)5 - 3\cdot6\cdot2 - (-2)1\cdot1 \\ = 4 + 15 + 24 + 20 - 36 + 2 = 29. \end{aligned}$$

There is a visual mnemonic for remembering how to compute the six terms in formula (7.1.11) for the determinant of 3×3 matrices. Write the matrix as a 3×5 array by repeating the first two columns, as shown in bold face in Figure 23: Then add the product of terms connected by solid lines sloping down and to the right and subtract the products of terms connected by dashed lines sloping up and to the right. Warning: this nice crisscross algorithm for computing determinants of 3×3 matrices does not generalize to $n \times n$ matrices.

When computing determinants of $n \times n$ matrices when $n > 3$, it is usually more efficient to compute the determinant using row reduction rather than by using formula (7.1.9). In the appendix to this chapter, Section 7.4, we verify that formula (7.1.9) actually satisfies the three properties of a determinant, thus completing the proof of Theorem 7.1.2.

An interesting and useful formula for reducing the effort in computing determinants is given by the following formula.

Lemma 7.1.9. *Let A be an $n \times n$ matrix of the form*

$$A = \begin{pmatrix} B & 0 \\ C & D \end{pmatrix},$$

where B is a $k \times k$ matrix and D is an $(n-k) \times (n-k)$ matrix. Then

$$\det(A) = \det(B)\det(D).$$

Proof We prove this result using (7.1.9) coupled with induction. Assume that this lemma is valid or all $(n-1) \times (n-1)$ matrices of the appropriate form. Now use (7.1.9) to compute

$$\begin{aligned} \det(A) &= a_{11}\det(A_{11}) - a_{12}\det(A_{12}) + \cdots \pm a_{1n}\det(A_{1n}) \\ &= b_{11}\det(A_{11}) - b_{12}\det(A_{12}) + \cdots \pm b_{1k}\det(A_{1k}). \end{aligned}$$

Note that the cofactor matrices A_{1j} are obtained from A by deleting the 1^{st} row and the j^{th} column. These

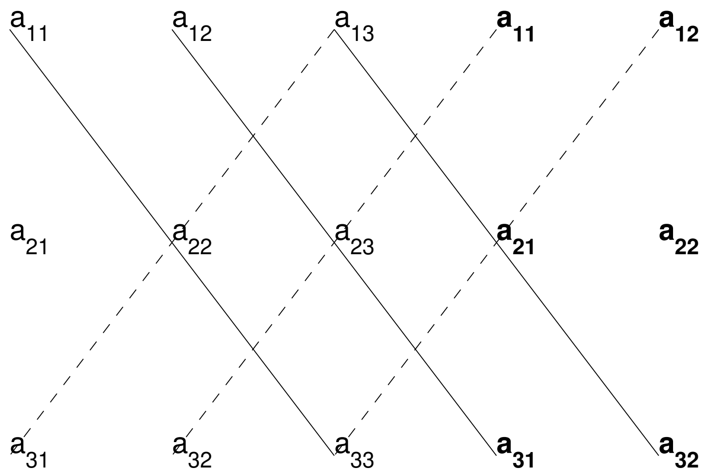

Figure 23: Mnemonic for computation of determinants of 3×3 matrices.

matrices all have the form

$$A_{1j} = \begin{pmatrix} B_{1j} & 0 \\ C_j & D \end{pmatrix},$$

where C_j is obtained from C by deleting the j^{th} column. By induction on k

$$\det(A_{1j}) = \det(B_{1j})\det(D).$$

It follows that

$$\begin{aligned}\det(A) &= (b_{11}\det(B_{11}) - b_{12}\det(B_{12}) + \cdots \\ &\qquad \pm b_{1k}\det(B_{1k}))\det(D) \\ &= \det(B)\det(D),\end{aligned}$$

as desired. ■

Determinants in MATLAB The determinant function has been preprogrammed in MATLAB and is quite easy to use. For example, typing `e8_1_11` will load the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 & 0 \\ 2 & 1 & 4 & 1 \\ -2 & -1 & 0 & 1 \\ -1 & 0 & -2 & 3 \end{pmatrix}. \qquad (7.1.12^*)$$

To compute the determinant of A just type `det(A)` and obtain the answer

```
ans =
   -46
```

Alternatively, we can use row reduction techniques in MATLAB to compute the determinant of A — just to test the theory that we have developed. Note that to compute the determinant we do not need to row reduce to reduced echelon form — we need only reduce to an upper triangular matrix. This can always be done by successively adding multiples of one row to another — an operation that does not change the determinant. For example, to clear the entries in the 1^{st} column below the 1^{st} row, type

```
A(2,:) = A(2,:) - 2*A(1,:);
A(3,:) = A(3,:) + 2*A(1,:);
A(4,:) = A(4,:) + A(1,:)
```

obtaining

```
A =
     1     2     3     0
     0    -3    -2     1
     0     3     6     1
     0     2     1     3
```

To clear the 2^{nd} column below the 2^{nd} row type

```
A(3,:) = A(3,:) + A(2,:);A(4,:)
= A(4,:) - A(4,2)*A(2,:)/A(2,2)
```

obtaining

```
A =
    1.0000    2.0000    3.0000         0
         0   -3.0000   -2.0000    1.0000
         0         0    4.0000    2.0000
         0         0   -0.3333    3.6667
```

Finally, to clear the entry $(4,3)$ type

```
A(4,:) = A(4,:) -A(4,3)*A(3,:)/A(3,3)
```

to obtain

```
A =
    1.0000    2.0000    3.0000         0
         0   -3.0000   -2.0000    1.0000
         0         0    4.0000    2.0000
         0         0         0    3.8333
```

To evaluate the determinant of A, which is now an upper triangular matrix, type

```
A(1,1)*A(2,2)*A(3,3)*A(4,4)
```

obtaining

```
ans =
   -46
```

as expected.

Exercises

7.2 Eigenvalues and Eigenvectors

In this section we discuss how to find eigenvalues for an $n \times n$ matrix A. This discussion parallels the discussion for 2×2 matrices given in Section 4.6. As we noted in that section, λ is a real eigenvalue of A if there exists a nonzero eigenvector v such that

$$Av = \lambda v. \tag{7.2.1}$$

It follows that the matrix $A - \lambda I_n$ is singular since

$$(A - \lambda I_n)v = 0.$$

Theorem 7.1.7 implies that

$$\det(A - \lambda I_n) = 0.$$

With these observations in mind, we can make the following definition.

Definition 7.2.1. Let A be an $n \times n$ matrix. The *characteristic polynomial* of A is:

$$p_A(\lambda) = \det(A - \lambda I_n).$$

In Theorem 7.2.3 we show that $p_A(\lambda)$ is indeed a polynomial of degree n in λ. Note here that the roots of p_A are the *eigenvalues* of A. As we discussed, the real eigenvalues of A are roots of the characteristic polynomial. Conversely, if λ is a real root of p_A, then Theorem 7.1.7 states that the matrix $A - \lambda I_n$ is singular and therefore that there exists a nonzero vector v such that (7.2.1) is satisfied. Similarly, by using this extended algebraic definition of eigenvalues we allow the possibility of complex eigenvalues. The complex analog of Theorem 7.1.7 shows that if λ is a complex eigenvalue, then there exists a nonzero complex n-vector v such that (7.2.1) is satisfied.

Example 7.2.2. *Let A be an $n \times n$ lower triangular matrix. Then the diagonal entries are the eigenvalues of A.* We verify this statement as follows.

$$A - \lambda I_n = \begin{pmatrix} a_{11} - \lambda & & 0 \\ & \ddots & \\ (*) & & a_{nn} - \lambda \end{pmatrix}.$$

Since the determinant of a triangular matrix is the product of the diagonal entries, it follows that

$$p_A(\lambda) = (a_{11} - \lambda) \cdots (a_{nn} - \lambda), \tag{7.2.2}$$

and hence that the diagonal entries of A are roots of the characteristic polynomial. A similar argument works if A is upper triangular.

It follows from (7.2.2) that the characteristic polynomial of a triangular matrix is a polynomial of degree n and that

$$p_A(\lambda) = (-1)^n \lambda^n + b_{n-1}\lambda^{n-1} + \cdots + b_0. \tag{7.2.3}$$

for some real constants $b_0, \ldots, b_{n-1}$. In fact, this statement is true in general.

Theorem 7.2.3. *Let A be an $n \times n$ matrix. Then p_A is a polynomial of degree n of the form (7.2.3).*

Proof Let C be an $n \times n$ matrix whose entries have the form $c_{ij} + d_{ij}\lambda$. Then $\det(C)$ is a polynomial in λ of degree at most n. We verify this statement by induction. It is easily verified when $n = 1$, since then $C = (c + d\lambda)$ for some real numbers c and d. Then $\det(C) = c + d\lambda$ which is a polynomial of degree at most one. (It may have degree zero, if $d = 0$.) So assume that this statement is true for $(n-1) \times (n-1)$ matrices. Recall from (7.1.9) that

$$\det(C) = (c_{11} + d_{11}\lambda)\det(C_{11}) + \cdots + (-1)^{n+1}(c_{1n} + d_{1n}\lambda)\det(C_{1n}).$$

By induction each of the determinants C_{1j} is a polynomial of degree at most $n-1$. It follows that multiplication by $c_{1j}+d_{1j}\lambda$ yields a polynomial of degree at most n in λ. Since the sum of polynomials of degree at most n is a polynomial of degree at most n, we have verified our assertion.

Since $A-\lambda I_n$ is a matrix whose entries have the desired form, it follows that $p_A(\lambda)$ is a polynomial of degree at most n in λ. To complete the proof of this theorem we need to show that the coefficient of λ^n is $(-1)^n$. Again, we verify this statement by induction. This statement is easily verified for 1×1 matrices — we assume that it is true for $(n-1)\times(n-1)$ matrices. Again use (7.1.9) to compute

$$\begin{aligned}\det(A-\lambda I_n) = (a_{11}-\lambda)\det(B_{11}) - a_{12}\det(B_{12}) + \cdots \\ + (-1)^{n+1}a_{1n}\det(B_{1n}).\end{aligned}$$

where B_{1j} are the cofactor matrices of $A-\lambda I_n$. Using our previous observation all of the terms $\det(B_{1j})$ are polynomials of degree at most $n-1$. Thus, in this expansion, the only term that can contribute a term of degree n is:

$$-\lambda\det(B_{11}).$$

Note that the cofactor matrix B_{11} is the $(n-1)\times(n-1)$ matrix

$$B_{11} = A_{11} - \lambda I_{n-1},$$

where A_{11} is the first cofactor matrix of the matrix A. By induction, $\det(B_{11})$ is a polynomial of degree $n-1$ with leading term $(-1)^{n-1}\lambda^{n-1}$. Multiplying this polynomial by $-\lambda$ yields a polynomial of degree n with the correct leading term. ■

General Properties of Eigenvalues The *fundamental theorem of algebra* states that every polynomial of degree n has exactly n roots (counting multiplicity). For example, the quadratic formula shows that every quadratic polynomial has exactly two roots. In general, the proof of the fundamental theorem is not easy and is certainly beyond the limits of this course. Indeed, the difficulty in proving the *fundamental theorem of algebra* is in proving that a polynomial $p(\lambda)$ of degree $n>0$ has one (complex) root. Suppose that λ_0 is a root of $p(\lambda)$; that is, suppose that $p(\lambda_0)=0$. Then it follows that

$$p(\lambda) = (\lambda-\lambda_0)q(\lambda) \tag{7.2.4}$$

for some polynomial q of degree $n-1$. So once we know that p has a root, then we can argue by induction to prove that p has n roots. A linear algebra proof of (7.2.4) is sketched in Exercise **??**.

Recall that a polynomial need not have any real roots. For example, the polynomial $p(\lambda)=\lambda^2+1$ has no real roots, since $p(\lambda)>0$ for all real λ. This polynomial does have two complex roots $\pm i = \pm\sqrt{-1}$.

However, a polynomial with real coefficients has either real roots or complex roots that come in complex conjugate pairs. To verify this statement, we need to show that if λ_0 is a complex root of $p(\lambda)$, then so is $\overline{\lambda_0}$. We claim that

$$p(\overline{\lambda}) = \overline{p(\lambda)}.$$

To verify this point, suppose that

$$p(\lambda) = c_n\lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_0,$$

where each $c_j\in\mathbb{R}$. Then

$$\begin{aligned}\overline{p(\lambda)} &= \overline{c_n\lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_0} \\ &= c_n\overline{\lambda}^n + c_{n-1}\overline{\lambda}^{n-1} + \cdots + c_0 \\ &= p(\overline{\lambda})\end{aligned}$$

If λ_0 is a root of $p(\lambda)$, then

$$p(\overline{\lambda_0}) = \overline{p(\lambda_0)} = \overline{0} = 0.$$

Hence $\overline{\lambda_0}$ is also a root of p.

It follows that

Theorem 7.2.4. *Every (real) $n \times n$ matrix A has exactly n eigenvalues $\lambda_1, \ldots, \lambda_n$. These eigenvalues are either real or complex conjugate pairs. Moreover,*

(a) $p_A(\lambda) = (\lambda_1 - \lambda) \cdots (\lambda_n - \lambda)$,

(b) $\det(A) = \lambda_1 \cdots \lambda_n$.

Proof Since the characteristic polynomial p_A is a polynomial of degree n with real coefficients, the first part of the theorem follows from the preceding discussion. In particular, it follows from (7.2.4) that

$$p_A(\lambda) = c(\lambda_1 - \lambda) \cdots (\lambda_n - \lambda),$$

for some constant c. Formula (7.2.3) implies that $c = 1$ — which proves (a). Since $p_A(\lambda) = \det(A - \lambda I_n)$, it follows that $p_A(0) = \det(A)$. Thus (a) implies that $p_A(0) = \lambda_1 \cdots \lambda_n$, thus proving (b). ■

The eigenvalues of a matrix do not have to be different. For example, consider the extreme case of a strictly triangular matrix A. Example 7.2.2 shows that all of the eigenvalues of A are zero.

We now discuss certain properties of eigenvalues.

Corollary 7.2.5. *Let A be an $n \times n$ matrix. Then A is invertible if and only if zero is* not *an eigenvalue of A.*

Proof The proof follows from Theorem 7.1.7 and Theorem 7.2.4(b). ■

Lemma 7.2.6. *Let A be a singular $n \times n$ matrix. Then the null space of A is the span of all eigenvectors whose associated eigenvalue is zero.*

Proof An eigenvector v of A has eigenvalue zero if and only if

$$Av = 0.$$

This statement is valid if and only if v is in the null space of A. ■

Theorem 7.2.7. *Let A be an invertible $n \times n$ matrix with eigenvalues $\lambda_1, \cdots, \lambda_n$. Then the eigenvalues of A^{-1} are $\lambda_1^{-1}, \cdots, \lambda_n^{-1}$.*

Proof We claim that

$$p_A(\lambda) = (-1)^n \det(A) \lambda^n p_{A^{-1}}(\frac{1}{\lambda}).$$

It then follows that $\frac{1}{\lambda}$ is an eigenvalue for A^{-1} for each eigenvalue λ of A. This makes sense, since the eigenvalues of A are nonzero.

Compute:

$$\begin{aligned}
(-1)^n \det(A)\lambda^n p_{A^{-1}}(\frac{1}{\lambda}) &= (-\lambda)^n \det(A) \det(A^{-1} - \frac{1}{\lambda} I_n) \\
&= \det(-\lambda A) \det(A^{-1} - \frac{1}{\lambda} I_n) \\
&= \det(-\lambda A(A^{-1} - \frac{1}{\lambda} I_n)) \\
&= \det(A - \lambda I_n) \\
&= p_A(\lambda),
\end{aligned}$$

which verifies the claim. ■

Theorem 7.2.8. *Let A and B be similar $n \times n$ matrices. Then*

$$p_A = p_B,$$

and hence the eigenvalues of A and B are identical.

Proof Since B and A are similar, there exists an invertible $n\times n$ matrix S such that $B = S^{-1}AS$. It follows that

$$\begin{aligned}\det(B-\lambda I_n) &= \det(S^{-1}AS-\lambda I_n)\\ &= \det(S^{-1}(A-\lambda I_n)S) = \det(A-\lambda I_n),\end{aligned}$$

which verifies that $p_A = p_B$. ■

Recall that the *trace* of an $n\times n$ matrix A is the sum of the diagonal entries of A; that is

$$\mathrm{tr}(A) = a_{11} + \cdots + a_{nn}.$$

We state without proof the following theorem:

Theorem 7.2.9. *Let A be an $n\times n$ matrix with eigenvalues $\lambda_1,\ldots,\lambda_n$. Then*

$$\mathrm{tr}(A) = \lambda_1 + \cdots + \lambda_n.$$

It follows from Theorem 7.2.8 that the traces of similar matrices are equal.

Definition 7.2.10. Associated with each eigenvalue λ of the square matrix A is a vector subspace of $\mathbb{R}^n$. This subspace, called the *eigenspace* of λ, is the *null space($A-\lambda I_n$)* and consists of all eigenvectors associated with the eigenvalue λ.

MATLAB Calculations The commands for computing characteristic polynomials and eigenvalues of square matrices are straightforward in MATLAB . In particular, for an $n\times n$ matrix A, the MATLAB command `poly(A)` returns the coefficients of $(-1)^n p_A(\lambda)$.

For example, reload the 4×4 matrix A of (7.1.12*) by typing `e8_1_11`. The characteristic polynomial of A is found by typing

```
poly(A)
```

to obtain

```
ans =
    1.0000   -5.0000   15.0000  -10.0000  -46.0000
```

Thus the characteristic polynomial of A is:

$$p_A(\lambda) = \lambda^4 - 5\lambda^3 + 15\lambda^2 - 10\lambda - 46.$$

The eigenvalues of A are found by typing `eig(A)` and obtaining

```
ans =
  -1.2224
   1.6605 + 3.1958i
   1.6605 - 3.1958i
   2.9014
```

Thus A has two real eigenvalues and one complex conjugate pair of eigenvalues. Note that MATLAB has preprogrammed not only the algorithm for finding the characteristic polynomial, but also numerical routines for finding the roots of the characteristic polynomial.

The trace of A is found by typing `trace(A)` and obtaining

```
ans =
     5
```

Using the MATLAB command `sum` we can verify the statement of Theorem 7.2.9. Indeed `sum(v)` computes the sum of the components of the vector v and typing

```
sum(eig(A))
```

we obtain the answer `5.0000`, as expected.

Exercises

7.3 Real Diagonalizable Matrices

An $n \times n$ matrix is *real diagonalizable* if it is similar to a diagonal matrix. More precisely, an $n \times n$ matrix A is real diagonalizable if there exists an invertible $n \times n$ matrix S such that

$$D = S^{-1}AS$$

is a diagonal matrix. In this section we investigate when a matrix is diagonalizable. In this discussion we assume that all matrices have real entries.

We begin with the observation that not all matrices are real diagonalizable. We saw in Example 7.2.2 that the diagonal entries of the diagonal matrix D are the eigenvalues of D. Theorem 7.2.8 states that similar matrices have the same eigenvalues. Thus if a matrix is real diagonalizable, then it must have real eigenvalues. It follows, for example, that the 2×2 matrix

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

is not real diagonalizable, since its eigenvalues are $\pm i$.

However, even if a matrix A has real eigenvalues, it need not be diagonalizable. For example, the only matrix similar to the identity matrix I_n is the identity matrix itself. To verify this point, calculate

$$S^{-1}I_nS = S^{-1}S = I_n.$$

Suppose that A is a matrix all of whose eigenvalues are equal to 1. If A is similar to a diagonal matrix D, then D must have all of its eigenvalues equal to 1. Since the identity matrix is the only diagonal matrix with all eigenvalues equal to 1, $D = I_n$. So, if A is similar to a diagonal matrix, it must itself be the identity matrix. Consider, however, the 2×2 matrix

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

Since A is triangular, it follows that both eigenvalues of A are equal to 1. Since A is not the identity matrix, it cannot be diagonalizable. More generally, if N is a nonzero strictly upper triangular $n \times n$ matrix, then the matrix $I_n + N$ is not diagonalizable.

These examples show that complex eigenvalues are always obstructions to real diagonalization and multiple real eigenvalues are sometimes obstructions to diagonalization. Indeed,

Theorem 7.3.1. *Let A be an $n \times n$ matrix with n distinct real eigenvalues. Then A is real diagonalizable.*

There are two ideas in the proof of Theorem 7.3.1, and they are summarized in the following lemmas.

Lemma 7.3.2. *Let $\lambda_1, \ldots, \lambda_k$ be distinct real eigenvalues for an $n \times n$ matrix A. Let v_j be eigenvectors associated with the eigenvalue λ_j. Then $\{v_1, \ldots, v_k\}$ is a linearly independent set.*

Proof We prove the lemma by using induction on k. When $k = 1$ the proof is simple, since $v_1 \neq 0$. So we can assume that $\{v_1, \ldots, v_{k-1}\}$ is a linearly independent set.

Let $\alpha_1, \ldots, \alpha_k$ be scalars such that

$$\alpha_1 v_1 + \cdots + \alpha_k v_k = 0. \tag{7.3.1}$$

We must show that all $\alpha_j = 0$.

Begin by multiplying both sides of (7.3.1) by A, to obtain:

$$\begin{aligned} 0 &= A(\alpha_1 v_1 + \cdots + \alpha_k v_k) \\ &= \alpha_1 A v_1 + \cdots + \alpha_k A v_k \\ &= \alpha_1 \lambda_1 v_1 + \cdots + \alpha_k \lambda_k v_k. \end{aligned} \tag{7.3.2}$$

Now subtract λ_k times (7.3.1) from (7.3.2), to obtain:

$$\alpha_1(\lambda_1 - \lambda_k)v_1 + \cdots + \alpha_{k-1}(\lambda_{k-1} - \lambda_k)v_{k-1} = 0.$$

Since $\{v_1, \ldots, v_{k-1}\}$ is a linearly independent set, it follows that

$$\alpha_j(\lambda_j - \lambda_k) = 0,$$

for $j = 1, \ldots, k-1$. Since all of the eigenvalues are distinct, $\lambda_j - \lambda_k \neq 0$ and $\alpha_j = 0$ for $j = 1, \ldots, k-1$. Substituting this information into (7.3.1) yields $\alpha_k v_k = 0$. Since $v_k \neq 0$, α_k is also equal to zero. ■

Lemma 7.3.3. *Let A be an $n \times n$ matrix. Then A is real diagonalizable if and only if A has n real linearly independent eigenvectors.*

Proof Suppose that A has n linearly independent eigenvectors $v_1, \ldots, v_n$. Let $\lambda_1, \ldots, \lambda_n$ be the corresponding eigenvalues of A; that is, $Av_j = \lambda_j v_j$. Let $S = (v_1|\cdots|v_n)$ be the $n \times n$ matrix whose columns are the eigenvectors v_j. We claim that $D = S^{-1}AS$ is a diagonal matrix. Compute

$$\begin{aligned} D = S^{-1}AS = S^{-1}A(v_1|\cdots|v_n) &= S^{-1}(Av_1|\cdots|Av_n) \\ = S^{-1}(\lambda_1 v_1|\cdots|\lambda_n v_n). & \end{aligned}$$

It follows that

$$D = (\lambda_1 S^{-1}v_1|\cdots|\lambda_n S^{-1}v_n).$$

Note that

$$S^{-1}v_j = e_j,$$

since

$$Se_j = v_j.$$

Therefore,

$$D = (\lambda_1 e_1|\cdots|\lambda_n e_n)$$

is a diagonal matrix.

Conversely, suppose that A is a real diagonalizable matrix. Then there exists an invertible matrix S such that $D = S^{-1}AS$ is diagonal. Let $v_j = Se_j$. We claim that $\{v_1, \ldots, v_n\}$ is a linearly independent set of eigenvectors of A.

Since D is diagonal, $De_j = \lambda_j e_j$ for some real number λ_j. It follows that

$$Av_j = SDS^{-1}v_j = SDS^{-1}Se_j = SDe_j = \lambda_j Se_j = \lambda_j v_j.$$

So v_j is an eigenvector of A. Since the matrix S is invertible, its columns are linearly independent. Since the columns of S are v_j, the set $\{v_1, \ldots, v_n\}$ is a linearly independent set of eigenvectors of A, as claimed. ■

Proof of Theorem 7.3.1 Let $\lambda_1, \ldots, \lambda_n$ be the distinct eigenvalues of A and let $v_1, \ldots, v_n$ be the corresponding eigenvectors. Lemma 7.3.2 implies that $\{v_1, \ldots, v_n\}$ is a linearly independent set in $\mathbb{R}^n$ and therefore a basis. Lemma 7.3.3 implies that A is diagonalizable. ■

Remark. Theorem 7.3.1 can be generalized as follows. Suppose all eigenvalues of the $n \times n$ matrix A are real. Then A is diagonalizable if and only if the dimension of the eigenspace associated with each eigenvalue λ is equal to the number of times λ is an eigenvalue of A. Issues surrounding this remark are discussed in Chapter 11.

Diagonalization Using MATLAB Let

$$A = \begin{pmatrix} -6 & 12 & 4 \\ 8 & -21 & -8 \\ -29 & 72 & 27 \end{pmatrix}. \tag{7.3.3*}$$

We use MATLAB to answer the questions: Is A real diagonalizable and, if it is, can we find the matrix S such that $S^{-1}AS$ is diagonal? We can find the eigenvalues of A by typing `eig(A)`. MATLAB's response is:

```
ans =
   -2.0000
   -1.0000
    3.0000
```

Since the eigenvalues of A are real and distinct, Theorem 7.3.1 states that A can be diagonalized. That is, there is a matrix S such that

$$S^{-1}AS = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

The proof of Lemma 7.3.3 tells us how to find the matrix S. We need to find the eigenvectors v_1, v_2, v_3 associated with the eigenvalues $-1, -2, 3$, respectively. Then the matrix $(v_1|v_2|v_3)$ whose columns are the eigenvectors is the matrix S. To verify this construction we first find the eigenvectors of A by typing

```
v1 = null(A+eye(3));
v2 = null(A+2*eye(3));
v3 = null(A-3*eye(3));
```

Now type `S = [v1 v2 v3]` to obtain

```
S =
    0.8729    0.7071         0
    0.4364    0.0000    0.3162
   -0.2182    0.7071   -0.9487
```

Finally, check that $S^{-1}AS$ is the desired diagonal matrix by typing `inv(S)*A*S` to obtain

```
ans =
   -1.0000    0.0000         0
    0.0000   -2.0000   -0.0000
    0.0000         0    3.0000
```

It is cumbersome to use the `null` command to find eigenvectors and MATLAB has been preprogrammed to do these computations automatically. We can use the `eig` command to find the eigenvectors and eigenvalues of a matrix A, as follows. Type

```
[S,D] = eig(A)
```

and MATLAB responds with

```
S =
   -0.7071    0.8729   -0.0000
   -0.0000    0.4364   -0.3162
   -0.7071   -0.2182    0.9487

D =
   -2.0000         0         0
         0   -1.0000         0
         0         0    3.0000
```

The matrix S is the transition matrix whose columns are the eigenvectors of A and the matrix D is a diagonal matrix whose j^{th} diagonal entry is the eigenvalue of A corresponding to the eigenvector in the j^{th} column of S.

Exercises

7.4 *Existence of Determinants

The purpose of this appendix is to verify the inductive definition of determinant (7.1.9). We have already shown that if a determinant function exists, then it is unique. We also know that the determinant function exists for 1×1 matrices. So we assume by induction that the determinant function exists for $(n-1) \times (n-1)$ matrices and prove that the inductive definition gives a determinant function for $n \times n$ matrices.

Recall that A_{ij} is the cofactor matrix obtained from A by deleting the i^{th} row and j^{th} column — so A_{ij} is an $(n-1) \times (n-1)$ matrix. The inductive definition is:

$$D(A) = a_{11}\det(A_{11}) - a_{12}\det(A_{12}) + \cdots + (-1)^{n+1} a_{1n}\det(A_{1n}).$$

We use the notation $D(A)$ to remind us that we have not yet verified that this definition satisfies properties (a)-(c) of Definition 7.1.1. In this appendix we verify these properties after assuming that the inductive definition satisfies properties (a)-(c) for $(n-1) \times (n-1)$ matrices. For emphasis, we use the notation det to indicate the determinant of square matrices of size less than n. Note that Lemma 7.1.5, the computation of determinants of elementary row operations, can therefore be assumed valid for $(n-1) \times (n-1)$ matrices.

We begin with the following two lemmas.

Lemma 7.4.1. *Let C be an $n \times n$ matrix. If two rows of C are equal or one row of C is zero, then $D(C) = 0$.*

Proof Suppose that row i of C is zero. If $i > 1$, then each cofactor has a zero row and by induction the determinant of the cofactor is 0. If row 1 is zero, then the cofactor expansion is 0 and $D(C) = 0$.

Suppose that row i and row j are equal, where $i > 1$ and $j > 1$. Then the result follows by the induction hypothesis, since each of the cofactors has two equal rows. So, we can assume that row 1 and row j are equal. If $j > 2$, let $\hat{C}$ be obtained from C by swapping rows j and 2. The cofactors $\hat{C}_{1k}$ are then obtained from the cofactors C_{1k} by swapping rows $j-1$ and 1. The induction hypothesis then implies that $\det(\hat{C}_{1k}) = -\det(C_{1k})$ and $\det(\hat{C}) = -\det(C)$. Thus, verifying that $\det(C) = 0$ reduces to verifying the result when rows 1 and 2 are equal.

Indeed, the most difficult part of this proof is the calculation that shows that if the 1^{st} and 2^{nd} rows of C are equal, then $D(C) = 0$. This calculation is tedious and requires some facility with indexes and summations. Rather than prove this for general n, we present the proof for $n = 4$. This case contains all of the ideas of the general proof.

We can assume that

$$C = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{pmatrix}$$

Using the cofactor definition $D(C) =$

$$a_1 \det\begin{pmatrix} a_2 & a_3 & a_4 \\ c_{32} & c_{33} & c_{34} \\ c_{42} & c_{43} & c_{44} \end{pmatrix} - a_2 \det\begin{pmatrix} a_1 & a_3 & a_4 \\ c_{31} & c_{33} & c_{34} \\ c_{41} & c_{43} & c_{44} \end{pmatrix} +$$
$$a_3 \det\begin{pmatrix} a_1 & a_2 & a_4 \\ c_{31} & c_{32} & c_{34} \\ c_{41} & c_{42} & c_{44} \end{pmatrix} - a_4 \det\begin{pmatrix} a_1 & a_2 & a_3 \\ c_{31} & c_{32} & c_{33} \\ c_{41} & c_{42} & c_{43} \end{pmatrix}.$$

Next we expand each of the four 3×3 matrices along their 1^{st} rows, obtaining $D(C) =$

$$a_1 \left(a_2 \det\begin{pmatrix} c_{33} & c_{34} \\ c_{43} & c_{44} \end{pmatrix} - a_3 \det\begin{pmatrix} c_{32} & c_{34} \\ c_{42} & c_{44} \end{pmatrix} + a_4 \det\begin{pmatrix} c_{32} & c_{33} \\ c_{42} & c_{43} \end{pmatrix} \right.$$
$$-a_2 \left(a_1 \det\begin{pmatrix} c_{33} & c_{34} \\ c_{43} & c_{44} \end{pmatrix} - a_3 \det\begin{pmatrix} c_{31} & c_{34} \\ c_{41} & c_{44} \end{pmatrix} + a_4 \det\begin{pmatrix} c_{31} & c_{3} \\ c_{41} & c_{4} \end{pmatrix} \right.$$
$$+a_3 \left(a_1 \det\begin{pmatrix} c_{32} & c_{34} \\ c_{42} & c_{44} \end{pmatrix} - a_2 \det\begin{pmatrix} c_{31} & c_{34} \\ c_{41} & c_{44} \end{pmatrix} + a_4 \det\begin{pmatrix} c_{31} & c_{3} \\ c_{41} & c_{4} \end{pmatrix} \right.$$
$$-a_4 \left(a_1 \det\begin{pmatrix} c_{32} & c_{33} \\ c_{42} & c_{43} \end{pmatrix} - a_2 \det\begin{pmatrix} c_{31} & c_{33} \\ c_{41} & c_{43} \end{pmatrix} + a_3 \det\begin{pmatrix} c_{31} & c_{3} \\ c_{41} & c_{4} \end{pmatrix} \right.$$

Combining the 2×2 determinants leads to $D(C) = 0$. ■

Lemma 7.4.2. *Let E be an $n\times n$ elementary row matrix and let B be an $n \times n$ matrix. Then*

$$D(EB) = D(E)D(B) \tag{7.4.1}$$

Proof We recall that the three elementary row operations are generated by two: (I) multiply row i by a nonzero scalar c and (II) add row i to row j. The remaining elementary row operations are obtained as follows. Adding c times row i to row j is the composition of multiplying row i by c, adding row i to row j, and multiplying row i by $1/c$. For 2×2 matrices swapping rows 1 and 2 was written in terms of four other elementary row operations in (7.1.4). This observation works in general, as follows. Consider the sequence of row operations:

- add row j to row i
- multiply row j by -1
- add row i to row j
- subtract row j from row i

We can write swapping rows i and j schematically as:

$$\begin{pmatrix} r_i \\ r_j \end{pmatrix} \to \begin{pmatrix} r_i + r_j \\ r_j \end{pmatrix} \to \begin{pmatrix} r_i + r_j \\ -r_j \end{pmatrix} \to \begin{pmatrix} r_i + r_j \\ r_i \end{pmatrix} \to \begin{pmatrix} r_j \\ r_i \end{pmatrix}$$

Thus, we need to verify (7.4.1) for two types of elementary row operation: multiply row i by $c\neq 0$ and add row j to row i.

(I) Suppose that E multiplies the i^{th} row by a nonzero scalar c. If $i > 1$, then the cofactor matrix $(EA)_{1j}$ is obtained from the cofactor matrix A_{1j} by multiplying the $(i-1)^{st}$ row by c. By induction, $\det(EA)_{1j} = c\det(A_{1j})$ and $D(EA) = cD(A)$. On the other hand, $D(E) = \det(E_{11}) = c$. So (7.4.1) is verified in this instance. If $i = 1$, then the 1^{st} row of EA is $(ca_{11},\ldots,ca_{1n})$ from which it is easy to use the cofactor formula to verify (7.4.1).

(II) Next suppose that E adds row i to row j. If $i, j > 1$, then the result follows from the induction hypothesis since the new cofactors are obtained from the old cofactors by adding row $i-1$ to row $j-1$.

If $j = 1$, then

$$\begin{aligned} D(EB) &= (b_{11}+b_{i1})\det(B_{11}) + \cdots + \\ &\quad (-1)^{n+1}(b_{1n}+b_{in})\det(B_{1n}) \\ &= \left[b_{11}\det(B_{11}) + \cdots + (-1)^{n+1}b_{1n}\det(B_{1n})\right] + \\ &\quad \left[b_{i1}\det(B_{11}) + \cdots + (-1)^{n+1}b_{i1}\det(B_{1n})\right] \\ &= D(B) + D(C) \end{aligned}$$

where the 1^{st} and i^{th} rows of C are equal. The fact that $D(C) = 0$ follows from Lemma 7.4.1.

If $i = 1$, then the cofactors are unchanged. It follows by direct calculation of the cofactor expansion that $D(EB) = D(B) + D(C)$ where the 1^{st} and i^{th} rows of C are equal. Again, the fact that $D(C) = 0$ follows from Lemma 7.4.1. ■

Property (a) is verified for $D(A)$ using cofactors since if A is lower triangular, then

$$D(A) = a_{11}\det(A_{11})$$

and

$$\det(A_{11}) = a_{22}\cdots a_{nn}$$

by the induction hypothesis.

Property (c) $(D(AB) = D(A)D(B))$ is proved separately for A singular and A nonsingular. In either case, row reduction implies that $A = E_s\cdots E_1R$ where R is in reduced echelon form.

If A is singular, the the bottom row of R is zero and together Lemmas 7.4.1 and 7.4.2 imply that $D(A) = 0$. On the other hand these lemmas also imply that

$$D(AB) = D(E_s\cdots E_1RB) = D(E_s\cdots E_1)D(RB)$$

and direct calculation shows that the bottom row of RB is also zero. Hence $D(RB) = 0$ and property (c) is valid.

Next suppose now that A is nonsingular. It follows that

$$AB = E_s \cdots E_1 B.$$

Using (7.4.1) we see that

$$D(AB) = D(E_s) \cdots D(E_1) D(B) = D(E_s \cdots E_1) D(B) = D(A) D(B),$$

as desired.

Before verifying property (b) we prove the following:

Lemma 7.4.3. *Let E be an elementary row operation matrix. Then $D(E^t) = D(E)$. An $n \times n$ matrix A is singular if and only if A^t is singular.*

Proof The two generators of elementary row operations are: multiply row i by c and add row i to row j. The first matrix is diagonal; so $E^t = E$. Denote the second matrix by F_{ij}. It follows that $F_{ij}^t = F_{ji}$. We claim that $D(F_{ij}) = 1$ for all i, j and hence that $D(E^t) = D(E)$ for all E. If $i < j$, then F_{ij} is lower triangular with 1^s on the diagonal. Hence $D(F_{ij}) = 1$. If $1 < j < i$, then $D(F_{ij}^t) = D(F_{ji}) = 1$ by induction. If $j = 1$, then $D(F_{i1}^t) = 1$ by direct calculation.

If A is singular, then $A = E_s \cdots E_1 R$, where R is in reduced echelon form and its bottom row is zero. Hence R is singular. It follows that $D(A) = 0$. Note that

$$A^t = R^t E_1^t \cdots E_s^t$$

Here we use the fact that $(BC)^t = C^t B^t$ that was discussed in (3.6.1). By counting pivots in R, we see that the column space and the row space of R have the same dimensions. Therefore, the dimension of the row space of R^t equals the dimension of the column space of R^t equals the dimension of the row space of R, and all of these are less than n. Hence R^t is singular. Therefore, there exists a nonzero n-vector w such that $R^t w = 0$. It follows that $v = (E_s^t)^{-1} \cdots (E_1^t)^{-1} w$ satisfies $A^t w = R^t v = 0$ and A^t is singular. ■

Property (b) We prove $D(A^t) = D(A)$ in two steps. Write

$$A = E_s \cdots E_1 R, \tag{7.4.2}$$

where the E_j are elementary row matrices and R is in reduced echelon form. It follows that

$$A^t = R^t E_1^t \cdots E_s^t. \tag{7.4.3}$$

If A is invertible, then $R = I_n$ and $D(A^t) = D(A)$. If A is singular, then A^t is also singular and $D(A) = 0 = D(A^t)$.

We have now completed the proof that a determinant function exists.

8 Linear Maps and Changes of Coordinates

The first section in this chapter, Section 8.1, defines linear mappings between abstract vector spaces, shows how such mappings are determined by their values on a basis, and derives basic properties of invertible linear mappings.

The notions of *row rank* and *column rank* of a matrix are discussed in Section 8.2 along with the theorem that states that these numbers are equal to the rank of that matrix.

Section 8.3 discusses the underlying meaning of similarity — the different ways to view the same linear mapping on $\mathbb{R}^n$ in different coordinates systems or bases. This discussion makes sense only after the definitions of coordinates corresponding to bases and of changes in coordinates are given and justified. In Section 8.4, we discuss the matrix associated to a linearity transformation between two finite dimensional vector spaces in a given set of coordinates and show that changes in coordinates correspond to similarity of the corresponding matrices.

8.1 Linear Mappings and Bases

The examples of linear mappings from $\mathbb{R}^n \to \mathbb{R}^m$ that we introduced in Section 3.3 were matrix mappings. More precisely, let A be an $m \times n$ matrix. Then

$$L_A(x) = Ax$$

defines the linear mapping $L_A : \mathbb{R}^n \to \mathbb{R}^m$. Recall that Ae_j is the j^{th} column of A (see Chapter 3, Lemma 3.3.4); it follows that A can be reconstructed from the vectors $Ae_1, \ldots, Ae_n$. This remark implies (Chapter 3, Lemma 3.3.3) that linear mappings of $\mathbb{R}^n$ to $\mathbb{R}^m$ are determined by their values on the standard basis $e_1, \ldots, e_n$. Next we show that this result is valid in greater generality. We begin by defining what we mean for a mapping between vector spaces to be linear.

Definition 8.1.1. Let V and W be vector spaces and let $L : V \to W$ be a mapping. The map L is *linear* if

$$\begin{aligned} L(u+v) &= L(u) + L(v) \\ L(cv) &= cL(v) \end{aligned}$$

for all $u, v \in V$ and $c \in \mathbb{R}$.

Examples of Linear Mappings (a) Let $v \in \mathbb{R}^n$ be a fixed vector. Use the dot product to define the mapping $L : \mathbb{R}^n \to \mathbb{R}$ by

$$L(x) = x \cdot v.$$

Then L is linear. Just check that

$$L(x+y) = (x+y) \cdot v = x \cdot v + y \cdot v = L(x) + L(y)$$

for every vector x and y in $\mathbb{R}^n$ and

$$L(cx) = (cx) \cdot v = c(x \cdot v) = cL(x)$$

for every scalar $c \in \mathbb{R}$.

(b) The map $L : \mathcal{C}^1 \to \mathbb{R}$ defined by

$$L(f) = f'(2)$$

is linear. Indeed,

$$L(f+g) = (f+g)'(2) = f'(2) + g'(2) = L(f) + L(g).$$

Similarly, $L(cf) = cL(f)$.

(c) The map $L : \mathcal{C}^1 \to \mathcal{C}^1$ defined by

$$L(f)(t) = f(t-1)$$

is linear. Indeed,

$$L(f+g)(t) = (f+g)(t-1) = f(t-1)+g(t-1) = L(f)(t)+L(g)(t).$$

Similarly, $L(cf) = cL(f)$. It may be helpful to compute $L(f)(t)$ when $f(t) = t^2 - t + 1$. That is,

$$L(f)(t) = (t-1)^2-(t-1)+1 = t^2-2t+1-t+1+1 = t^2-3t+3.$$

Constructing Linear Mappings from Bases

Theorem 8.1.2. *Let V and W be vector spaces. Let $\{v_1, \ldots, v_n\}$ be a basis for V and let $\{w_1, \ldots, w_n\}$ be n vectors in W. Then there exists a unique linear map $L : V \to W$ such that $L(v_i) = w_i$.*

Proof Let $v \in V$ be a vector. Since $\mathrm{span}\{v_1, \ldots, v_n\} = V$, we may write v as

$$v = \alpha_1 v_1 + \cdots + \alpha_n v_n,$$

where $\alpha_1, \ldots, \alpha_n$ in $\mathbb{R}$. Moreover, $v_1, \ldots, v_n$ are linearly independent, these scalars are uniquely defined. More precisely, if

$$\alpha_1 v_1 + \cdots + \alpha_n v_n = \beta_1 v_1 + \cdots + \beta_n v_n,$$

then

$$(\alpha_1 - \beta_1)v_1 + \cdots + (\alpha_n - \beta_n)v_n = 0.$$

Linear independence implies that $\alpha_j - \beta_j = 0$; that is $\alpha_j = \beta_j$. We can now define

$$L(v) = \alpha_1 w_1 + \cdots + \alpha_n w_n. \tag{8.1.1}$$

We claim that L is linear. Let $\hat{v} \in V$ be another vector and let

$$\hat{v} = \beta_1 v_1 + \cdots + \beta_n v_n.$$

It follows that

$$v + \hat{v} = (\alpha_1 + \beta_1)v_1 + \cdots + (\alpha_n + \beta_n)v_n,$$

and hence by (8.1.1) that

$$\begin{aligned} L(v+\hat{v}) &= (\alpha_1 + \beta_1)w_1 + \cdots + (\alpha_n + \beta_n)w_n \\ &= (\alpha_1 w_1 + \cdots + \alpha_n w_n) + (\beta_1 w_1 + \cdots + \beta_n w_n) \\ &= L(v) + L(\hat{v}). \end{aligned}$$

Similarly

$$\begin{aligned} L(cv) &= L((c\alpha_1)v_1 + \cdots + (c\alpha_n)v_n) \\ &= c(\alpha_1 w_1 + \cdots + \alpha_n w_n) \\ &= cL(v). \end{aligned}$$

Thus L is linear.

Let $M : V \to W$ be another linear mapping such that $M(v_i) = w_i$. Then

$$\begin{aligned} L(v) &= L(\alpha_1 v_1 + \ldots + \alpha_n v_n) \\ &= \alpha_1 w_1 + \cdots + \alpha_n w_n \\ &= \alpha_1 M(v_1) + \cdots + \alpha_n M(v_n) \\ &= M(\alpha_1 v_1 + \cdots + \alpha_n v_n) \\ &= M(v). \end{aligned}$$

Thus $L = M$ and the linear mapping is uniquely defined. ■

There are two assertions made in Theorem 8.1.2. The first is that a linear map exists mapping v_i to w_i. The second is that there is only one *linear* mapping that accomplishes this task. If we drop the constraint that the map be linear, then many mappings may satisfy these conditions. For example, find a linear map from $\mathbb{R} \to \mathbb{R}$ that maps 1 to 4. There is only one: $y = 4x$. However there are many nonlinear maps that send 1 to 4. Examples are $y = x + 3$ and $y = 4x^2$.

Finding the Matrix of a Linear Map from $\mathbb{R}^n \to \mathbb{R}^m$ Given by Theorem 8.1.2 Suppose that $V = \mathbb{R}^n$ and $W = \mathbb{R}^m$. We know that every linear map $L : \mathbb{R}^n \to \mathbb{R}^m$ can be defined as multiplication by an $m \times n$ matrix. The question that we next address is: How can we find the matrix whose existence is guaranteed by Theorem 8.1.2?

More precisely, let $v_1, \ldots, v_n$ be a basis for $\mathbb{R}^n$ and let $w_1, \ldots, w_n$ be vectors in $\mathbb{R}^m$. We suppose that all of these vectors are row vectors. Then we need to find an $m \times n$ matrix A such that $Av_i^t = w_i^t$ for all i. We find A as follows. Let $v \in \mathbb{R}^n$ be a row vector. Since the v_i form a basis, there exist scalars α_i such that

$$v = \alpha_1 v_1 + \cdots + \alpha_n v_n.$$

In coordinates

$$v^t = (v_1^t | \cdots | v_n^t) \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}, \tag{8.1.2}$$

where $(v_1^t | \cdots | v_n^t)$ is an $n \times n$ invertible matrix. By definition (see (8.1.1))

$$L(v) = \alpha_1 w_1 + \cdots + \alpha_n w_n.$$

Thus the matrix A must satisfy

$$Av^t = (w_1^t | \cdots | w_n^t) \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix},$$

where $(w_1^t|\cdots|w_n^t)$ is an $m\times n$ matrix. Using (8.1.2) we see that

$$Av^t = (w_1^t|\cdots|w_n^t)(v_1^t|\cdots|v_n^t)^{-1}v^t,$$

and

$$A = (w_1^t|\cdots|w_n^t)(v_1^t|\cdots|v_n^t)^{-1} \tag{8.1.3}$$

is the desired $m\times n$ matrix.

An Example of a Linear Map from $\mathbb{R}^3$ to $\mathbb{R}^2$ As an example we illustrate Theorem 8.1.2 and (8.1.3) by defining a linear mapping from $\mathbb{R}^3$ to $\mathbb{R}^2$ by its action on a basis. Let

$$v_1 = (1,4,1) \quad v_2 = (-1,1,1) \quad v_3 = (0,1,0).$$

We claim that $\{v_1,v_2,v_3\}$ is a basis of $\mathbb{R}^3$ and that there is a unique linear map for which $L(v_i)=w_i$ where

$$w_1 = (2,0) \quad w_2 = (1,1) \quad w_3 = (1,-1).$$

We can verify that $\{v_1,v_2,v_3\}$ is a basis of $\mathbb{R}^3$ by showing that the matrix

$$(v_1^t|v_2^t|v_3^t) = \begin{pmatrix} 1 & -1 & 0\\ 4 & 1 & 1\\ 1 & 1 & 0\end{pmatrix}$$

is invertible. This can either be done in MATLAB using the `inv` command or by hand by row reducing the matrix

$$\left(\begin{array}{ccc|ccc} 1 & -1 & 0 & 1 & 0 & 0\\ 4 & 1 & 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 0 & 0 & 1\end{array}\right)$$

to obtain

$$(v_1^t|v_2^t|v_3^t)^{-1} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 1\\ -1 & 0 & 1\\ -3 & 2 & -5\end{pmatrix}.$$

Now apply (8.1.3) to obtain

$$A = \frac{1}{2}\begin{pmatrix} 2 & 1 & 1\\ 0 & 1 & -1\end{pmatrix}\begin{pmatrix} 1 & 0 & 1\\ -1 & 0 & 1\\ -3 & 2 & -5\end{pmatrix} = \begin{pmatrix} -1 & 1 & -1\\ 1 & -1 & 3\end{pmatrix}.$$

As a check, verify by matrix multiplication that $Av_i = w_i$, as claimed.

Properties of Linear Mappings

Lemma 8.1.3. *Let U,V,W be vector spaces and $L:V\to W$ and $M:U\to V$ be linear maps. Then $L\circ M:U\to W$ is linear.*

Proof The proof of Lemma 8.1.3 is identical to that of Chapter 3, Lemma 3.5.1. ■

A linear map $L:V\to W$ is *invertible* if there exists a linear map $M:W\to V$ such that $L\circ M:W\to W$ is the identity map on W and $M\circ L:V\to V$ is the identity map on V.

Theorem 8.1.4. *Let V and W be finite dimensional vector spaces and let $v_1,\ldots,v_n$ be a basis for V. Let $L:V\to W$ be a linear map. Then L is invertible if and only if $w_1,\ldots,w_n$ is a basis for W where $w_j = L(v_j)$.*

Proof If $w_1,\ldots,w_n$ is a basis for W, then use Theorem 8.1.2 to define a linear map $M:W\to V$ by $M(w_j)=v_j$. Note that

$$L\circ M(w_j) = L(v_j) = w_j.$$

It follows by linearity (using the uniqueness part of Theorem 8.1.2) that $L\circ M$ is the identity of W. Similarly, $M\circ L$ is the identity map on V, and L is invertible.

Conversely, suppose that $L\circ M$ and $M\circ L$ are identity maps and that $w_j = L(v_j)$. We must show that

$w_1, \ldots, w_n$ is a basis. We use Theorem 5.5.3 and verify separately that $w_1, \ldots, w_n$ are linearly independent and span W.

If there exist scalars $\alpha_1, \ldots, \alpha_n$ such that

$$\alpha_1 w_1 + \cdots + \alpha_n w_n = 0,$$

then apply M to both sides of this equation to obtain

$$0 = M(\alpha_1 w_1 + \cdots + \alpha_n w_n) = \alpha_1 v_1 + \cdots + \alpha_n v_n.$$

But the v_j are linearly independent. Therefore, $\alpha_j = 0$ and the w_j are linearly independent.

To show that the w_j span W, let w be a vector in W. Since the v_j are a basis for V, there exist scalars $\beta_1, \ldots, \beta_n$ such that

$$M(w) = \beta_1 v_1 + \cdots + \beta_n v_n.$$

Applying L to both sides of this equation yields

$$w = L \circ M(w) = \beta_1 w_1 + \cdots + \beta_n w_n.$$

Therefore, the w_j span W. ■

Corollary 8.1.5. *Let V and W be finite dimensional vector spaces. Then there exists an invertible linear map $L : V \to W$ if and only if $\dim(V) = \dim(W)$.*

Proof Suppose that $L : V \to W$ is an invertible linear map. Let $v_1, \ldots, v_n$ be a basis for V where $n = \dim(V)$. Then Theorem 8.1.4 implies that $L(v_1), \ldots, L(v_n)$ is a basis for W and $\dim(W) = n = \dim(V)$.

Conversely, suppose that $\dim(V) = \dim(W) = n$. Let $v_1, \ldots, v_n$ be a basis for V and let $w_1, \ldots, w_n$ be a basis for W. Using Theorem 8.1.2 define the linear map $L : V \to W$ by $L(v_j) = w_j$. Theorem 8.1.4 states that L is invertible. ■

Exercises

8.2 Row Rank Equals Column Rank

Let A be an $m \times n$ matrix. The *row space* of A is the span of the row vectors of A and is a subspace of $\mathbb{R}^n$. The *column space* of A is the span of the columns of A and is a subspace of $\mathbb{R}^m$.

Definition 8.2.1. The *row rank* of A is the dimension of the row space of A and the *column rank* of A is the dimension of the column space of A.

Lemma 5.5.4 of Chapter 5 states that

$$\text{row rank}(A) = \text{rank}(A).$$

We show below that row ranks and column ranks are equal. We begin by continuing the discussion of the previous section on linear maps between vector spaces.

Null Space and Range Each linear map between vector spaces defines two subspaces. Let V and W be vector spaces and let $L : V \to W$ be a linear map. Then

$$\text{null space}(L) = \{v \in V : L(v) = 0\} \subset V$$

and

$$\text{range}(L) = \{L(v) \in W : v \in V\} \subset W.$$

Lemma 8.2.2. *Let $L : V \to W$ be a linear map between vector spaces. Then the null space of L is a subspace of V and the range of L is a subspace of W.*

Proof The proof that the null space of L is a subspace of V follows from linearity in precisely the same way that the null space of an $m \times n$ matrix is a subspace of $\mathbb{R}^n$. That is, if v_1 and v_2 are in the null space of L, then

$$L(v_1 + v_2) = L(v_1) + L(v_2) = 0 + 0 = 0,$$

and for $c \in \mathbb{R}$

$$L(cv_1) = cL(v_1) = c0 = 0.$$

So the null space of L is closed under addition and scalar multiplication and is a subspace of V.

To prove that the range of L is a subspace of W, let w_1 and w_2 be in the range of L. Then, by definition, there exist v_1 and v_2 in V such that $L(v_j) = w_j$. It follows that

$$L(v_1 + v_2) = L(v_1) + L(v_2) = w_1 + w_2.$$

Therefore, $w_1 + w_2$ is in the range of L. Similarly,

$$L(cv_1) = cL(v_1) = cw_1.$$

So the range of L is closed under addition and scalar multiplication and is a subspace of W. ■

Suppose that A is an $m \times n$ matrix and $L_A : \mathbb{R}^n \to \mathbb{R}^m$ is the associated linear map. Then the null space of L_A is precisely the null space of A, as defined in Definition 5.2.1 of Chapter 5. Moreover, the range of L_A is the column space of A. To verify this, write $A = (A_1|\cdots|A_n)$ where A_j is the j^{th} column of A and let $v = (v_1, \ldots v_n)^t$. Then, $L_A(v)$ is the linear combination of columns of A

$$L_A(v) = Av = v_1A_1 + \cdots + v_nA_n.$$

There is a theorem that relates the dimensions of the null space and range with the dimension of V.

Theorem 8.2.3. *Let V and W be vector spaces with V finite dimensional and let $L : V \to W$ be a linear map. Then*

$$\dim(V) = \dim(\text{null space}(L)) + \dim(\text{range}(L)).$$

Proof Since V is finite dimensional, the null space of L is finite dimensional (since the null space is a subspace of V) and the range of L is finite dimensional (since it is spanned by the vectors $L(v_j)$ where $v_1, \ldots, v_n$ is a basis for V). Let $u_1, \ldots, u_k$ be a basis for the null space of L and let $w_1, \ldots, w_\ell$ be a basis for the range of L. Choose vectors $y_j \in V$ such that $L(y_j) = w_j$. We claim that $u_1, \ldots, u_k, y_1, \ldots, y_\ell$ is a basis for V, which proves the theorem.

To verify that $u_1, \ldots, u_k, y_1, \ldots, y_\ell$ are linear independent, suppose that

$$\alpha_1 u_1 + \cdots + \alpha_k u_k + \beta_1 y_1 + \cdots + \beta_\ell y_\ell = 0. \tag{8.2.1}$$

Apply L to both sides of (8.2.1) to obtain

$$\beta_1 w_1 + \cdots + \beta_\ell w_\ell = 0.$$

Since the w_j are linearly independent, it follows that $\beta_j = 0$ for all j. Now (8.2.1) implies that

$$\alpha_1 u_1 + \cdots + \alpha_k u_k = 0.$$

Since the u_j are linearly independent, it follows that $\alpha_j = 0$ for all j.

To verify that $u_1, \ldots, u_k, y_1, \ldots, y_\ell$ span V, let v be in V. Since $w_1, \ldots, w_\ell$ span W, it follows that there exist scalars β_j such that

$$L(v) = \beta_1 w_1 + \cdots + \beta_\ell w_\ell.$$

Note that by choice of the y_j

$$L(\beta_1 y_1 + \cdots + \beta_\ell y_\ell) = \beta_1 w_1 + \cdots + \beta_\ell w_\ell.$$

It follows by linearity that

$$u = v - (\beta_1 y_1 + \cdots + \beta_\ell y_\ell)$$

is in the null space of L. Hence there exist scalars α_j such that

$$u = \alpha_1 u_1 + \cdots + \alpha_k u_k.$$

Thus, v is in the span of $u_1, \ldots, u_k, y_1, \ldots, y_\ell$, as desired. ∎

Row Rank and Column Rank Recall Theorem 5.5.6 of Chapter 5 that states that the nullity plus the rank of an $m \times n$ matrix equals n. At first glance it might seem that this theorem and Theorem 8.2.3 contain the same information, but they do not. Theorem 5.5.6 of Chapter 5 is proved using a detailed analysis of solutions of linear equations based on Gaussian elimination, back substitution, and reduced echelon form, while Theorem 8.2.3 is proved using abstract properties of linear maps.

Let A be an $m \times n$ matrix. Theorem 5.5.6 of Chapter 5 states that

$$\mathrm{nullity}(A) + \mathrm{rank}(A) = n.$$

Meanwhile, Theorem 8.2.3 states that

$$\dim(\text{null space}(L_A)) + \dim(\text{range}(L_A)) = n.$$

But the dimension of the null space of L_A equals the nullity of A and the dimension of the range of A equals the dimension of the column space of A. Therefore,

$$\mathrm{nullity}(A) + \dim(\text{column space}(A)) = n.$$

Hence, the rank of A equals the column rank of A. Since rank and row rank are identical, we have proved:

Theorem 8.2.4. *Let A be an $m \times n$ matrix. Then*

$$\textit{row rank } A = \textit{column rank } A.$$

Since the row rank of A equals the column rank of A^t, we have:

Corollary 8.2.5. *Let A be an $m \times n$ matrix. Then*

$$\mathrm{rank}(A) = \mathrm{rank}(A^t).$$

Exercises

8.3 Vectors and Matrices in Coordinates

In the last half of this chapter we discuss how similarity of matrices should be thought of as change of coordinates for linear mappings. There are three steps in this discussion.

(a) Formalize the idea of coordinates for a vector in terms of basis.

(b) Discuss how to write a linear map as a matrix in each coordinate system.

(c) Determine how the matrices corresponding to the same linear map in two different coordinate systems are related.

The answer to the last question is simple: the matrices are related by a change of coordinates if and only if they are similar. We discuss these steps in this section in $\mathbb{R}^n$ and in Section 8.4 for general vector spaces.

Coordinates of Vectors using Bases Throughout, we have written vectors $v \in \mathbb{R}^n$ in coordinates as $v = (v_1, \ldots, v_n)$, and we have used this notation almost without comment. From the point of view of vector space operations, we are just writing

$$v = v_1e_1 + \cdots + v_ne_n$$

as a linear combination of the standard basis $\mathcal{E} = \{e_1, \ldots, e_n\}$ of $\mathbb{R}^n$.

More generally, each basis provides a set of coordinates for a vector space. This fact is described by the following lemma (although its proof is identical to the first part of the proof of Theorem 8.1.2.

Lemma 8.3.1. *Let $\mathcal{W} = \{w_1, \ldots, w_n\}$ be a basis for the vector space V. Then each vector v in V can be written uniquely as a linear combination of vectors in $\mathcal{W}$; that is,*

$$v = \alpha_1w_1 + \cdots + \alpha_nw_n,$$

for uniquely defined scalars $\alpha_1, \ldots, \alpha_n$.

Proof Since $\mathcal{W}$ is a basis, Theorem 5.5.3 of Chapter 5 implies that the vectors $w_1, \ldots, w_n$ span V and are linearly independent. Therefore, we can write v in V as a linear combination of vectors in $\mathcal{B}$. That is, there are scalars $\alpha_1, \ldots, \alpha_n$ such that

$$v = \alpha_1w_1 + \cdots + \alpha_nw_n.$$

Next we show that these scalars are uniquely defined. Suppose that we can write v as a linear combination of the vectors in $\mathcal{B}$ in a second way; that is, suppose

$$v = \beta_1w_1 + \cdots + \beta_nw_n$$

for scalars $\beta_1, \ldots, \beta_n$. Then

$$(\alpha_1 - \beta_1)w_1 + \cdots + (\alpha_n - \beta_n)w_n = 0.$$

Since the vectors in $\mathcal{W}$ are linearly independent, it follows that $\alpha_j = \beta_j$ for all j. ■

Definition 8.3.2. Let $\mathcal{W} = \{w_1, \ldots, w_n\}$ be a basis in a vector space V. Lemma 8.3.1 states that we can write $v \in V$ uniquely as

$$v = \alpha_1w_1 + \cdots + \alpha_nw_n. \tag{8.3.1}$$

The scalars $\alpha_1, \ldots, \alpha_n$ are the *coordinates* of v relative to the basis $\mathcal{W}$, and we denote the coordinates of v in the basis $\mathcal{W}$ by

$$[v]_{\mathcal{W}} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n. \tag{8.3.2}$$

We call the coordinates of a vector $v \in \mathbb{R}^n$ relative to the standard basis, the *standard coordinates* of v.

Writing Linear Maps in Coordinates as Matrices Let V be a finite dimensional vector space of dimension n and let $L : V \to V$ be a linear mapping. We now show how each basis of V allows us to associate an $n \times n$ matrix to L. Previously we considered this question with the standard basis on $V = \mathbb{R}^n$. We showed in Chapter 3 that we can write the linear mapping L as a matrix mapping, as follows. Let $\mathcal{E} = \{e_1, \ldots, e_n\}$ be the standard basis in $\mathbb{R}^n$. Let A be the $n \times n$ matrix whose j^{th} column is the n vector $L(e_j)$. Then Chapter 3, Theorem 3.3.5 shows that the linear map is given by matrix multiplication as

$$L(v) = Av.$$

Thus every linear mapping on $\mathbb{R}^n$ can be written in this matrix form.

Remark. Another way to think of the j^{th} column of the matrix A is as the coordinate vector of $L(e_j)$ relative to the standard basis, that is, as $[L(e_j)]_{\mathcal{E}}$. We denote the matrix A by $[L]_{\mathcal{E}}$; this notation emphasizes the fact that A is the matrix of L relative to the standard basis.

We now discuss how to write a linear map L as a matrix using different coordinates.

Definition 8.3.3. Let $\mathcal{W} = \{w_1, \ldots, w_n\}$ be a basis for the vector space V. The $n \times n$ matrix $[L]_{\mathcal{W}}$ associated to the linear map $L : V \to V$ and the basis $\mathcal{W}$ is defined as follows. The j^{th} column of $[L]_{\mathcal{W}}$ is $[L(w_j)]_{\mathcal{W}}$ — the coordinates of $L(w_j)$ relative to the basis $\mathcal{W}$.

Note that when $V = \mathbb{R}^n$ and when $\mathcal{W} = \mathcal{E}$, the standard basis of $\mathbb{R}^n$, then the definition of the matrix $[L]_{\mathcal{E}}$ is exactly the same as the matrix associated with the linear map L in Remark 8.3.

Lemma 8.3.4. *The coordinate vector of $L(v)$ relative to the basis $\mathcal{W}$ is*

$$[L(v)]_{\mathcal{W}} = [L]_{\mathcal{W}}[v]_{\mathcal{W}}. \qquad (8.3.3)$$

Proof The process of choosing the coordinates of vectors relative to a given basis $\mathcal{W} = \{w_1, \ldots, w_n\}$ of a vector space V is itself linear. Indeed,

$$\begin{aligned} [u+v]_{\mathcal{W}} &= [u]_{\mathcal{W}} + [v]_{\mathcal{W}} \\ [cv]_{\mathcal{W}} &= c[v]_{\mathcal{W}}. \end{aligned}$$

Thus the coordinate mapping relative to a basis $\mathcal{W}$ of V defined by

$$v \mapsto [v]_{\mathcal{W}} \qquad (8.3.4)$$

is a linear mapping of V into $\mathbb{R}^n$. We denote this linear mapping by $[\cdot]_{\mathcal{W}} : V \to \mathbb{R}^n$.

It now follows that both the left hand and right hand sides of (8.3.3) can be thought of as linear mappings of $V \to \mathbb{R}^n$. In verifying this comment, we recall Lemma 8.1.3 of Chapter 5 that states that the composition of linear maps is linear. On the left hand side we have the mapping

$$v \mapsto L(v) \mapsto [L(v)]_{\mathcal{W}},$$

which is the composition of the linear maps: $[\cdot]_{\mathcal{W}}$ with L. See (8.3.4). The right hand side is

$$v \mapsto [v]_{\mathcal{W}} \mapsto [L]_{\mathcal{W}}[v]_{\mathcal{W}},$$

which is the composition of the linear maps: multiplication by the matrix $[L]_{\mathcal{W}}$ with $[\cdot]_{\mathcal{W}}$.

Theorem 8.1.2 states that linear mappings are determined by their actions on a basis. Thus to verify (8.3.3), we need only verify this equality for $v = w_j$ for all j. Since $[w_j]_{\mathcal{W}} = e_j$, the right hand side of (8.3.3) is:

$$[L]_{\mathcal{W}}[w_j]_{\mathcal{W}} = [L]_{\mathcal{W}}e_j,$$

which is just the j^{th} column of $[L]_{\mathcal{W}}$. The left hand side of (8.3.3) is the vector $[L(w_j)]_{\mathcal{W}}$, which by definition is also the j^{th} column of $[L]_{\mathcal{W}}$ (see Definition 8.3.3). ∎

Computations of Vectors in Coordinates in $\mathbb{R}^n$ We divide this subsection into three parts. We consider a simple example in $\mathbb{R}^2$ algebraically in the first part and geometrically in the second. In the third part we formalize and extend the algebraic discussion to $\mathbb{R}^n$.

An Example of Coordinates in $\mathbb{R}^2$ How do we find the coordinates of a vector v in a basis? For example, choose a (nonstandard) basis in the plane — say

$$w_1 = (1,1) \quad \text{and} \quad w_2 = (1,-2).$$

Since $\{w_1, w_2\}$ is a basis, we may write the vector v as a linear combination of the vectors w_1 and w_2. Thus we can find scalars α_1 and α_2 so that

$$v = \alpha_1 w_1 + \alpha_2 w_2 = \alpha_1(1,1) + \alpha_2(1,-2) = (\alpha_1+\alpha_2, \alpha_1-2\alpha_2).$$

In standard coordinates, set $v = (v_1, v_2)$; this equation leads to the system of linear equations

$$\begin{array}{rcl} v_1 & = & \alpha_1 + \alpha_2 \\ v_2 & = & \alpha_1 - 2\alpha_2 \end{array}$$

in the two variables α_1 and α_2. As we have seen, the fact that w_1 and w_2 form a basis of $\mathbb{R}^2$ implies that these equations do have a solution. Indeed, we can write this system in matrix form as

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix},$$

which is solved by inverting the matrix to obtain:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \tag{8.3.5}$$

For example, suppose $v = (2.0, 0.5)$. Using (8.3.5) we find that $(\alpha_1, \alpha_2) = (1.5, 0.5)$; that is, we can write

$$v = 1.5w_1 + 0.5w_2,$$

and $(1.5, 0.5)$ are the *coordinates* of v in the basis $\{w_1, w_2\}$.

Using the notation in (8.3.2), we may rewrite (8.3.5) as

$$[v]_{\mathcal{W}} = \frac{1}{3} \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix} [v]_{\mathcal{E}},$$

where $\mathcal{E} = \{e_1, e_2\}$ is the standard basis.

Planar Coordinates Viewed Geometrically using MATLAB Next we use MATLAB to view geometrically the notion of coordinates relative to a basis $\mathcal{W} = \{w_1, w_2\}$ in the plane. Type

```
w1 = [1 1];
w2 = [1 -2];
bcoord
```

MATLAB will create a graphics window showing the two basis vectors w_1 and w_2 in red. Using the mouse click on a point near $(2, 0.5)$ in that figure. MATLAB will respond by plotting the new vector v in yellow and the parallelogram generated by $\alpha_1 w_1$ and $\alpha_2 w_2$ in cyan. The values of α_1 and α_2 are also plotted on this figure. See Figure 24.

Abstracting $\mathbb{R}^2$ to $\mathbb{R}^n$ Suppose that we are given a basis $\mathcal{W} = \{w_1, \ldots, w_n\}$ of $\mathbb{R}^n$ and a vector $v \in \mathbb{R}^n$. How do we find the coordinates $[v]_{\mathcal{W}}$ of v in the basis $\mathcal{W}$?

For definiteness, assume that v and the w_j are row vectors. Equation (8.3.1) may be rewritten as

$$v^t = (w_1^t | \cdots | w_n^t) \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}.$$

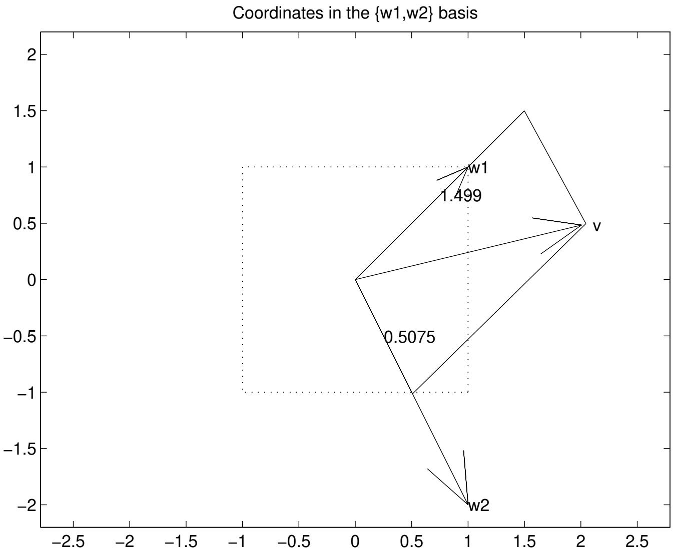

Figure 24: The coordinates of $v = (2.0, 0.5)$ in the basis $w_1 = (1, 1), w_2 = (1, -2)$.

Thus,

$$[v]_{\mathcal{W}} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = P_{\mathcal{W}}^{-1} v^t, \qquad (8.3.6)$$

where $P_{\mathcal{W}} = (w_1^t|\cdots|w_n^t)$. Since the w_j are a basis for $\mathbb{R}^n$, the columns of the matrix $P_{\mathcal{W}}$ are linearly independent, and $P_{\mathcal{W}}$ is invertible.

We may use (8.3.6) to compute $[v]_{\mathcal{W}}$ using MATLAB. For example, let

$$v = (4, 1, 3)$$

and

$$w_1 = (1, 4, 7) \quad w_2 = (2, 1, 0) \quad w_3 = (-4, 2, 1).$$

Then $[v]_{\mathcal{W}}$ is found by typing

```
w1 = [ 1 4 7];
w2 = [ 2 1 0];
w3 = [-4 2 1];
inv([w1' w2' w3'])*[4 1 3]'
```

The answer is:

```
ans =
    0.5306
    0.3061
   -0.7143
```

Determining the Matrix of a Linear Mapping in Coordinates Suppose that we are given the linear map $L_A : \mathbb{R}^n \to \mathbb{R}^n$ associated to the matrix A in standard coordinates and a basis $w_1, \ldots, w_n$ of $\mathbb{R}^n$. How do we find the matrix $[L_A]_{\mathcal{W}}$. As above, we assume that the vectors w_j and the vector v are row vectors Since $L_A(v) = Av^t$ we can rewrite (8.3.3) as

$$[L_A]_{\mathcal{W}}[v]_{\mathcal{W}} = [Av^t]_{\mathcal{W}}$$

As above, let $P_{\mathcal{W}} = (w_1^t|\cdots|w_n^t)$. Using (8.3.6) we see that

$$[L_A]_{\mathcal{W}} P_{\mathcal{W}}^{-1} v^t = P_{\mathcal{W}}^{-1} A v^t.$$

Setting

$$u = P_{\mathcal{W}}^{-1} v^t$$

we see that

$$[L_A]_{\mathcal{W}} u = P_{\mathcal{W}}^{-1} A P_{\mathcal{W}} u.$$

Therefore,

$$[L_A]_{\mathcal{W}} = P_{\mathcal{W}}^{-1} A P_{\mathcal{W}}.$$

We have proved:

Theorem 8.3.5. *Let A be an $n \times n$ matrix and let $L_A : \mathbb{R}^n \to \mathbb{R}^n$ be the associated linear map. Let $\mathcal{W} = \{w_1, \ldots, w_n\}$ be a basis for $\mathbb{R}^n$. Then the matrix $[L_A]_{\mathcal{W}}$ associated to to L_A in the basis $\mathcal{W}$ is similar to A. Therefore the determinant, trace, and eigenvalues of $[L_A]_{\mathcal{W}}$ are identical to those of A.*

Matrix Normal Forms in $\mathbb{R}^2$ If we are careful about how we choose the basis $\mathcal{W}$, then we can simplify the form of the matrix $[L]_{\mathcal{W}}$. Indeed, we have already seen examples of this process when we discussed how to find closed form solutions to linear planar systems of ODEs in the previous chapter. For example, suppose that $L : \mathbb{R}^2 \to \mathbb{R}^2$ has real eigenvalues λ_1 and λ_2 with two linearly independent eigenvectors w_1 and w_2. Then the matrix associated to L in the basis $\mathcal{W} = \{w_1, w_2\}$ is the diagonal matrix

$$[L]_{\mathcal{W}} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \tag{8.3.7}$$

since

$$[L(w_1)]_{\mathcal{W}} = [\lambda_1 w_1]_{\mathcal{W}} = \begin{pmatrix} \lambda_1 \\ 0 \end{pmatrix}$$

and

$$[L(w_2)]_{\mathcal{W}} = [\lambda_2 w_2]_{\mathcal{W}} = \begin{pmatrix} 0 \\ \lambda_2 \end{pmatrix}.$$

In Chapter 6 we showed how to classify 2×2 matrices up to similarity (see Theorem 6.3.4) and how to use this classification to find closed form solutions to planar systems of linear ODEs (see Section 6.3). We now use the ideas of coordinates and matrices associated with bases to reinterpret the normal form result (Theorem 6.3.4) in a more geometric fashion.

Theorem 8.3.6. *Let $L : \mathbb{R}^2 \to \mathbb{R}^2$ be a linear mapping. Then in an appropriate coordinate system defined by the basis $\mathcal{W}$ below, the matrix $[L]_{\mathcal{W}}$ has one of the following forms.*

(a) Suppose that L has two linearly independent real eigenvectors w_1 and w_2 with real eigenvalues λ_1 and λ_2. Then

$$[L]_{\mathcal{W}} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

(b) Suppose that L has no real eigenvectors and complex conjugate eigenvalues $\sigma \pm i\tau$ where $\tau \neq 0$. Let $w_1 + iw_2$ be a complex eigenvector of L associated with the eigenvalue $\sigma - i\tau$. Then $\mathcal{W} = \{w_1, w_2\}$ is a basis and

$$[L]_{\mathcal{W}} = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix}.$$

(c) Suppose that L has exactly one linearly independent real eigenvector w_1 with real eigenvalue λ. Choose the generalized eigenvector w_2

$$(L - \lambda I_2)(w_2) = w_1. \tag{8.3.8}$$

Then $\mathcal{W} = \{w_1, w_2\}$ is a basis and

$$[L]_{\mathcal{W}} = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}.$$

Proof The verification of (a) was discussed in (8.3.7). The verification of (b) follows from (6.2.9) on equating w_1 with v and w_2 with w. The verification of (c) follows directly from (8.3.8) as

$$[L(w_1)]_{\mathcal{W}} = \lambda e_1 \quad \text{and} \quad [L(w_2)]_{\mathcal{W}} = e_1 + \lambda e_2.$$

■

Visualization of Coordinate Changes in ODEs We consider two examples. As a first example note that the matrices

$$C = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4 & -3 \\ 6 & -5 \end{pmatrix},$$

are similar matrices. Indeed, $B = P^{-1}CP$ where

$$P = \begin{pmatrix} 2 & -1 \\ 1 & -1 \end{pmatrix}. \tag{8.3.9}$$

The phase portraits of the differential equations $\dot{X} = BX$ and $\dot{X} = CX$ are shown in Figure 25. Note that both

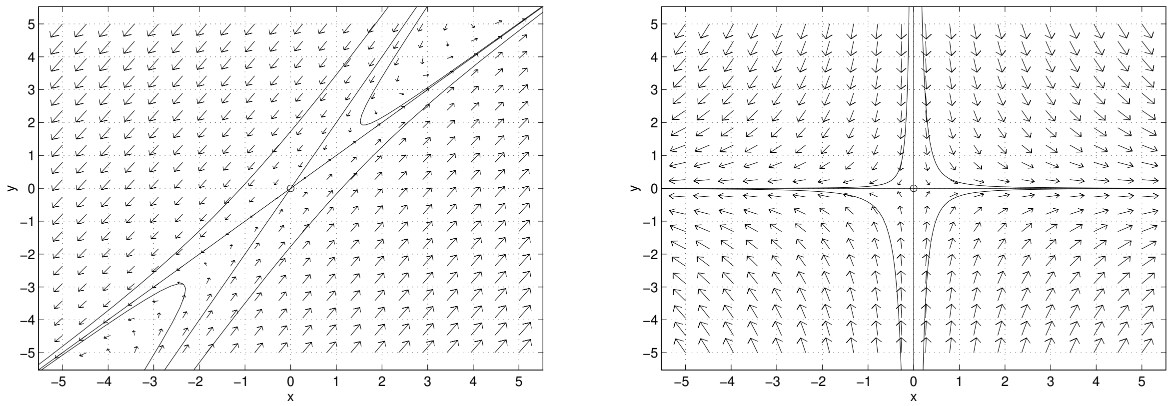

Figure 25: Phase planes for the saddles $\dot{X} = BX$ and $\dot{X} = CX$.

phase portraits are pictures of the *same* saddle — just in different coordinate systems.

As a second example note that the matrices

$$C = \begin{pmatrix} 0 & 2 \\ -2 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 6 & -4 \\ 10 & -6 \end{pmatrix}$$

are similar matrices, and both are centers. Indeed, $B = P^{-1}CP$ where P is the same matrix as in (8.3.9). The phase portraits of the differential equations $\dot{X} = BX$ and $\dot{X} = CX$ are shown in Figure 26. Note that both phase portraits are pictures of the *same* center — just in different coordinate systems.

Exercises

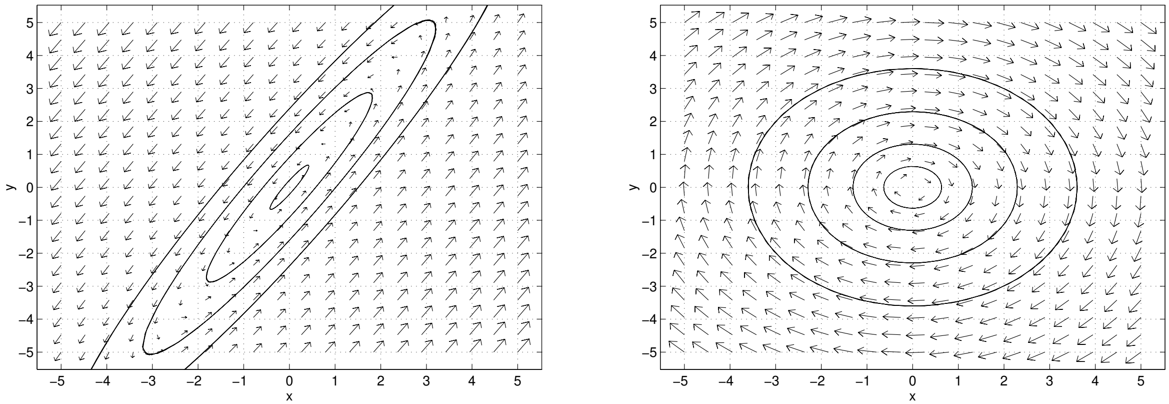

Figure 26: Phase planes for the centers $\dot{X} = BX$ and $\dot{X} = CX$.

8.4 *Matrices of Linear Maps on a Vector Space

Returning to the general finite dimensional vector space V, suppose that

$$\mathcal{W} = \{w_1, \ldots, w_n\} \quad \text{and} \quad \mathcal{Z} = \{z_1, \ldots, z_n\}$$

are bases of V. Then we can write

$$v = \alpha_1 w_1 + \cdots + \alpha_n w_n \quad \text{and} \quad v = \beta_1 z_1 + \cdots + \beta_n z_n$$

to obtain the coordinates

$$[v]_{\mathcal{W}} = (\alpha_1, \ldots, \alpha_n) \quad \text{and} \quad [v]_{\mathcal{Z}} = (\beta_1, \ldots, \beta_n) \quad (8.4.1)$$

of v relative to the bases $\mathcal{W}$ and $\mathcal{Z}$. The question that we address is: How are $[v]_{\mathcal{W}}$ and $[v]_{\mathcal{Z}}$ related? We answer this question by finding an $n \times n$ matrix $C_{\mathcal{WZ}}$ such that

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = C_{\mathcal{WZ}} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}. \quad (8.4.2)$$

We may rewrite (8.4.2) as

$$[v]_{\mathcal{W}} = C_{\mathcal{WZ}}[v]_{\mathcal{Z}}. \quad (8.4.3)$$

Definition 8.4.1. Let $\mathcal{W}$ and $\mathcal{Z}$ be bases for the n-dimensional vector space V. The $n \times n$ matrix $C_{\mathcal{WZ}}$ is a *transition* matrix if $C_{\mathcal{WZ}}$ satisfies (8.4.3).

Transition Mappings Defined The next theorem presents a method for finding the transition matrix between coordinates associated to bases in an n-dimensional vector space V.

Theorem 8.4.2. *Let $\mathcal{W} = \{w_1, \ldots, w_n\}$ and $\mathcal{Z} = \{z_1, \ldots, z_n\}$ be bases for the n-dimensional vector space V. Then*

$$C_{\mathcal{WZ}} = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \vdots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix} \quad (8.4.4)$$

is the transition matrix, where

$$\begin{aligned} z_1 &= c_{11}w_1 + \cdots + c_{n1}w_n \\ &\vdots \\ z_n &= c_{1n}w_1 + \cdots + c_{nn}w_n \end{aligned} \quad (8.4.5)$$

for scalars c_{ij}.

Proof We can restate (8.4.5) as

$$[z_j]_{\mathcal{W}} = \begin{pmatrix} c_{1j} \\ \vdots \\ c_{nj} \end{pmatrix}.$$

Note that

$$[z_j]_{\mathcal{Z}} = e_j,$$

by definition. Since the transition matrix satisfies $[v]_{\mathcal{W}} = C_{\mathcal{WZ}}[v]_{\mathcal{Z}}$ for all vectors $v \in V$, it must satisfy this relation for $v = z_j$. Therefore,

$$[z_j]_{\mathcal{W}} = C_{\mathcal{WZ}}[z_j]_{\mathcal{Z}} = C_{\mathcal{WZ}}e_j.$$

It follows that $[z_j]_{\mathcal{W}}$ is the j^{th} column of $C_{\mathcal{WZ}}$, which proves the theorem. ■

A Formula for $C_{\mathcal{WZ}}$ when $V = \mathbb{R}^n$ For bases in $\mathbb{R}^n$, there is a formula for finding transition matrices. Let $\mathcal{W} = \{w_1, \ldots, w_n\}$ and $\mathcal{Z} = \{z_1, \ldots, z_n\}$ be bases of $\mathbb{R}^n$ — written as row vectors. Also, let $v \in \mathbb{R}^n$ be written as a row vector. Then (8.3.6) implies that

$$[v]_{\mathcal{W}} = P_{\mathcal{W}}^{-1}v^t \quad \text{and} \quad [v]_{\mathcal{Z}} = P_{\mathcal{Z}}^{-1}v^t,$$

where

$$P_{\mathcal{W}} = (w_1^t|\cdots|w_n^t) \quad \text{and} \quad P_{\mathcal{Z}} = (z_1^t|\cdots|z_n^t).$$

It follows that

$$[v]_{\mathcal{W}} = P_{\mathcal{W}}^{-1}P_{\mathcal{Z}}[v]_{\mathcal{Z}}$$

and that

$$C_{\mathcal{W}\mathcal{Z}} = P_{\mathcal{W}}^{-1}P_{\mathcal{Z}}. \tag{8.4.6}$$

As an example, consider the following bases of $\mathbb{R}^4$. Let

$$\begin{array}{rclcrcl} w_1 & = & [1,4,2,3] & & z_1 & = & [3,2,0,1] \\ w_2 & = & [2,1,1,4] & & z_2 & = & [-1,0,2,3] \\ w_3 & = & [0,1,5,6] & & z_3 & = & [3,1,1,3] \\ w_4 & = & [2,5,-1,0] & & z_4 & = & [2,2,3,5] \end{array} \tag{8.4.7*}$$

Then the matrix $C_{\mathcal{W}\mathcal{Z}}$ is obtained by typing `e9_4_7` to enter the bases and

```
inv([w1' w2' w3' w4'])*[z1' z2' z3' z4']
```

to compute $C_{\mathcal{W}\mathcal{Z}}$. The answer is:

```
ans =
   -8.0000    5.5000   -7.0000   -3.2500
   -0.5000    0.7500    0.0000    0.1250
    4.5000   -2.7500    4.0000    2.3750
    6.0000   -4.0000    5.0000    2.5000
```

Coordinates Relative to Two Different Bases in $\mathbb{R}^2$ Recall the basis $\mathcal{W}$

$$w_1 = (1,1) \quad \text{and} \quad w_2 = (1,-2)$$

of $\mathbb{R}^2$ that was used in a previous example. Suppose that $\mathcal{Z} = \{z_1, z_2\}$ is a second basis of $\mathbb{R}^2$. Write $v = (v_1, v_2)$ as a linear combination of the basis $\mathcal{Z}$

$$v = \beta_1 z_1 + \beta_2 z_2,$$

obtaining the coordinates $[v]_{\mathcal{Z}} = (\beta_1, \beta_2)$.

We use MATLAB to illustrate how the coordinates of a vector v relative to two bases may be viewed geometrically. Suppose that $z_1 = (1,3)$ and $z_2 = (1,-2)$. Then enter the two bases $\mathcal{W}$ and $\mathcal{Z}$ by typing

```
w1 = [1 1];
w2 = [1 -2];
z1 = [1 3];
z2 = [-1 2];
ccoord
```

The MATLAB program `ccoord` opens two graphics windows representing the $\mathcal{W}$ and $\mathcal{Z}$ planes with the basis vectors plotted in red. Clicking the left mouse button on a vector in the $\mathcal{W}$ plane simultaneously plots this vector v in both planes in yellow and the coordinates of v in the respective bases in cyan. See Figure 27. From this display you can visualize the coordinates of a vector relative to two different bases.

Note that the program `ccoord` prints the transition matrix $C_{\mathcal{W}\mathcal{Z}}$ in the MATLAB control window. We can verify the calculations of the program `ccoord` on this example by hand. Recall that (8.4.6) states that

$$\begin{aligned} C_{\mathcal{W}\mathcal{Z}} &= \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 2 \\ 4 & 1 \end{pmatrix} \\ &= \begin{pmatrix} -3 & 2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 4 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 5 & -4 \\ -2 & 3 \end{pmatrix}. \end{aligned}$$

Matrices of Linear Maps in Different Bases

Theorem 8.4.3. *Let $L : V \to V$ be a linear mapping and let $\mathcal{W}$ and $\mathcal{Z}$ be bases of V. Then*

$$[L]_{\mathcal{Z}} \quad \textit{and} \quad [L]_{\mathcal{W}}$$

are similar matrices. More precisely,

$$[L]_{\mathcal{W}} = C_{\mathcal{Z}\mathcal{W}}^{-1}[L]_{\mathcal{Z}}C_{\mathcal{Z}\mathcal{W}}. \tag{8.4.8}$$

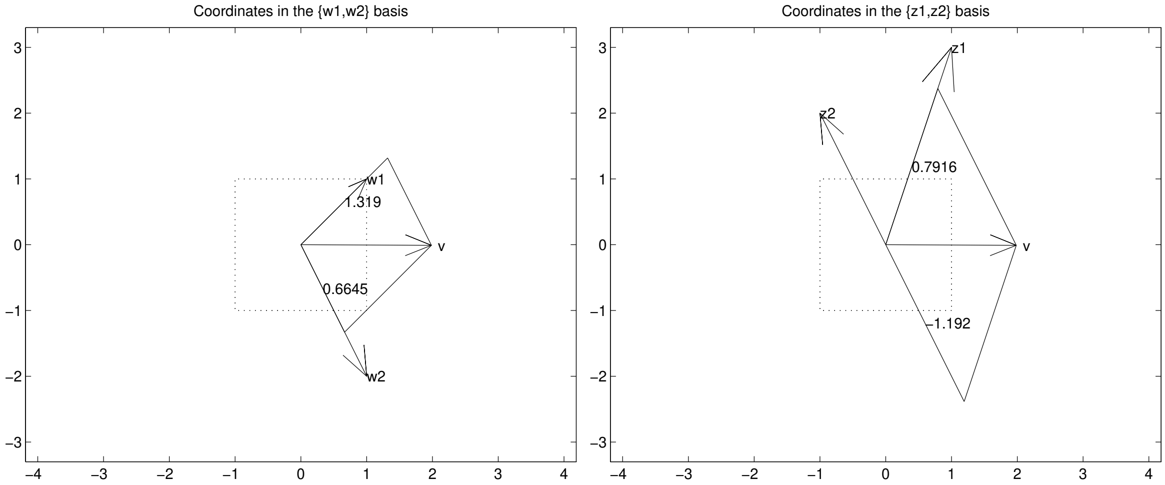

Figure 27: The coordinates of $v = (1.9839, -0.0097)$ in the bases $w_1 = (1,1), w_2 = (1,-2)$ and $z_1 = (1,3), z_2 = (-1,2)$.

Proof For every $v \in \mathbb{R}^n$ we compute

$$\begin{aligned} C_{\mathcal{ZW}}[L]_{\mathcal{W}}[v]_{\mathcal{W}} &= C_{\mathcal{ZW}}[L(v)]_{\mathcal{W}} \\ &= [L(v)]_{\mathcal{Z}} \\ &= [L]_{\mathcal{Z}}[v]_{\mathcal{Z}} \\ &= [L]_{\mathcal{Z}}C_{\mathcal{ZW}}[v]_{\mathcal{W}}. \end{aligned}$$

Since this computation holds for every $[v]_{\mathcal{W}}$, it follows that

$$C_{\mathcal{ZW}}[L]_{\mathcal{W}} = [L]_{\mathcal{Z}}C_{\mathcal{ZW}}.$$

Thus (8.4.8) is valid. ∎

Exercises

9 Least Squares

In Section 9.1 we study the geometric problem of least squares approximations: Given a point x_0 and a subspace $W \subset \mathbb{R}^n$, find the point $w_0 \in W$ closest to x_0. We then use least squares approximation to discuss regression or least squares fitting of data in Section 9.2.

9.1 Least Squares Approximations

Let $W \subset \mathbb{R}^n$ be a subspace and $x_0 \in \mathbb{R}^n$ be a vector. In this section we solve a basic geometric problem and investigate some of its consequences. The problem is:

> Find a vector $w_0 \in W$ that is the nearest vector in W to x_0.

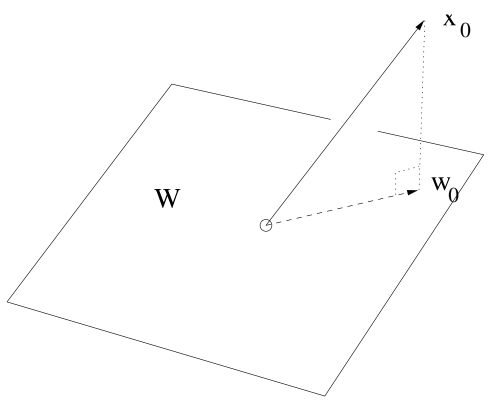

Figure 28: Approximation of x_0 by $w_0 \in W$ by least squares.

The distance between two vectors v and w is $||v - w||$ and the geometric problem can be rephrased as follows: find a vector $w_0 \in W$ such that

$$||x_0 - w_0|| \leq ||x_0 - w|| \quad \forall w \in W. \tag{9.1.1}$$

Condition (9.1.1) is called the *least squares approximation*. In order to see where this name comes from we write(9.1.1) in the equivalent form

$$||x_0 - w_0||^2 \leq ||x_0 - w||^2 \quad \forall w \in W.$$

This form means that for $w = w_0$ the sum of the squares of the components of the vector $x_0 - w$ is minimal.

Before continuing, we state and prove the *Law of Pythagorus*. Let $z_1, z_2 \in \mathbb{R}^n$ be orthogonal vectors. Then

$$||z_1 + z_2||^2 = ||z_1||^2 + ||z_2||^2. \tag{9.1.2}$$

To verify (9.1.2) calculate

$$\begin{aligned} ||z_1 + z_2||^2 &= (z_1 + z_2) \cdot (z_1 + z_2) \\ &= z_1 \cdot z_1 + 2z_1 \cdot z_2 + z_2 \cdot z_2 \\ &= ||z_1||^2 + 2z_1 \cdot z_2 + ||z_2||^2. \end{aligned}$$

Since z_1 and z_2 are orthogonal, $z_1 \cdot z_2 = 0$ and the Law of Pythagorus is valid.

Using (9.1.1) and (9.1.2), we can rephrase the minimum distance problem as follows.

Lemma 9.1.1. *The vector $w_0 \in W$ is the closest vector to $x_0 \in \mathbb{R}^n$ if the vector $x_0 - w_0$ is orthogonal to every vector in W. (See Figure 28.)*

Proof Write $x_0 - w = z_1 + z_2$ where $z_1 = x_0 - w_0$ and $z_2 = w_0 - w$. By assumption, $x_0 - w_0$ is orthogonal to every vector in W; so z_1 and $z_2 \in W$ are orthogonal. It follows from (9.1.2) that

$$||x_0 - w||^2 = ||x_0 - w_0||^2 + ||w_0 - w||^2.$$

Since $||w_0 - w||^2 \geq 0$, (9.1.1) is valid, and w_0 is the vector nearest to x_0 in W. ■

Least Squares Distance to a Line Suppose W is as simple a subspace as possible; that is, suppose W is one dimensional with basis vector w. Since W is one dimensional, a vector $w_0 \in W$ that is closest to x_0 must be a multiple of w; that is, $w_0 = aw$. Suppose that we can find a scalar a so that $x_0 - aw$ is orthogonal to every vector in W. Then it follows from Lemma 9.1.1 that w_0 is the closest vector in W to x_0. To find a, calculate

$$0 = (x_0 - aw) \cdot w = x_0 \cdot w - aw \cdot w.$$

Then

$$a = \frac{x_0 \cdot w}{||w||^2}$$

and

$$w_0 = \frac{x_0 \cdot w}{||w||^2} w. \tag{9.1.3}$$

Observe that $||w||^2 \neq 0$ since w is a basis vector.

For example, if $x_0 = (1, 2, -1, 3) \in \mathbb{R}^4$ and $w = (0, 1, 2, 3)$. The the vector w_0 in the space spanned by w that is nearest to x_0 is

$$w_0 = \frac{9}{14} w$$

since $x_0 \cdot w = 9$ and $||w||^2 = 14$.

Least Squares Distance to a Subspace Similarly, using Lemma 9.1.1 we can solve the general least squares problem by solving a system of linear equations. Let $w_1, \ldots, w_k$ be a basis for W and suppose that

$$w_0 = \alpha_1 w_1 + \cdots + \alpha_k w_k$$

for some scalars α_i. We now show how to find these scalars.

Theorem 9.1.2. *Let $x_0 \in \mathbb{R}^n$ be a vector, and let $\{w_1, \ldots, w_k\}$ be a basis for the subspace $W \subset \mathbb{R}^n$. Then*

$$w_0 = \alpha_1 w_1 + \cdots + \alpha_k w_k$$

is the nearest vector in W to x_0 when

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} = (A^t A)^{-1} A^t x_0, \tag{9.1.4}$$

where $A = (w_1 | \cdots | w_k)$ is the $n \times k$ matrix whose columns are the basis vectors of W.

Proof Observe that the vector $x_0 - w_0$ is orthogonal to every vector in W precisely when $x_0 - w_0$ is orthogonal to each basis vector w_j. It follows from Lemma 9.1.1 that w_0 is the closest vector to x_0 in W if

$$(x_0 - w_0) \cdot w_j = 0$$

for every j. That is, if

$$w_0 \cdot w_j = x_0 \cdot w_j$$

for every j. These equations can be rewritten as a system of equations in terms of the α_i, as follows:

$$\begin{array}{rcl} w_1 \cdot w_1 \alpha_1 + \cdots + w_1 \cdot w_k \alpha_k & = & w_1 \cdot x_0 \\ & \vdots & \\ w_k \cdot w_1 \alpha_1 + \cdots + w_k \cdot w_k \alpha_k & = & w_k \cdot x_0. \end{array} \tag{9.1.5}$$

Note that if $u, v \in \mathbb{R}^n$ are column vectors, then $u \cdot v = u^t v$. Therefore, we can rewrite (9.1.5) as

$$A^t A \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} = A^t x_0,$$

where A is the matrix whose columns are the w_j and x_0 is viewed as a column vector. Note that the matrix $A^t A$ is a $k \times k$ matrix.

We claim that $A^t A$ is invertible. To verify this claim, it suffices to show that the null space of $A^t A$ is zero; that is, if $A^t A z = 0$ for some $z \in \mathbb{R}^k$, then $z = 0$. First, calculate

$$||Az||^2 = Az \cdot Az = (Az)^t Az = z^t A^t A z = z^t 0 = 0.$$

It follows that $Az = 0$. Now if we let $z = (z_1, \ldots, z_k)^t$, then the equation $Az = 0$ may be rewritten as

$$z_1 w_1 + \cdots + z_k w_k = 0.$$

Since the w_j are linearly independent, it follows that the $z_j = 0$. In particular, $z = 0$. Since $A^t A$ is invertible, (9.1.4) is valid, and the theorem is proved. ■

Exercises

9.2 Least Squares Fitting of Data

We begin this section by using the method of least squares to find the best straight line fit to a set of data. Later in the section we will discuss best fits to other curves.

An Example of Best Linear Fit to Data Suppose that we are given n data points (x_i, y_i) for $i = 1, \ldots, 10$. For example, consider the ten points

$$\begin{array}{ccccc} (2.0, 0.1) & (3.0, 2.7) & (1.5, -1.1) & (-1.0, -5.5) & (0.0, -3.4) \\ (3.6, 3.0) & (0.7, -2.8) & (4.1, 4.0) & (1.9, -1.9) & (5.0, 5.5) \end{array}$$
(9.2.1*)

The ten points (x_i, y_i) are plotted in Figure 29 using the commands

```
e9_3_1
plot(X,Y,'o')
axis([-3,7,-8,8])
xlabel('x')
ylabel('y')
```

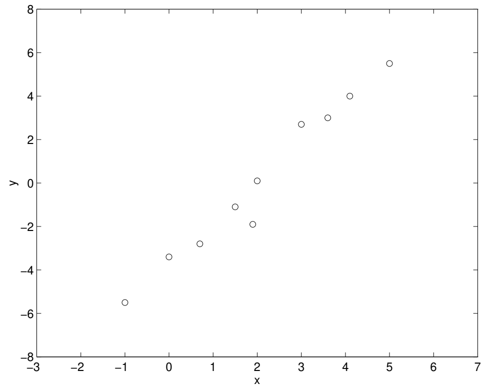

Figure 29: Scatter plot of data in (9.2.1*).

Next, suppose that there is a linear relation between the x_i and the y_i; that is, we assume that there are constants b_1 and b_2 (that do not depend on i) for which $y_i = b_1 + b_2 x_i$ for each i. But these points are just data; errors may have been made in their measurement. So we ask: Find b_1^0 and b_2^0 so that the error made in fitting the data to the line $y = b_1^0 + b_2^0 x$ is minimal, that is, the error that is made in that fit is less than or equal to the error made in fitting the data to the line $y = b_1 + b_2 x$ for any other choice of b_1 and b_2.

We begin by discussing what that error actually is. Given constants b_1 and b_2 and given a data point x_i, the difference between the data value y_i and the hypothesized value $b_1 + b_2 x_i$ is the error that is made at that data point. Next, we combine the errors made at all of the data points; a standard way to combine the errors is to use the Euclidean distance

$$E(b) = \left((y_1 - (b_1 + b_2 x_1))^2 + \cdots + (y_{10} - (b_1 + b_2 x_{10}))^2\right)^{\frac{1}{2}}.$$

Rewriting $E(b)$ in vector notation leads to an economy in notation and to a conceptual advantage. Let

$$X = (x_1, \ldots, x_{10})^t \quad Y = (y_1, \ldots, y_{10})^t \quad \text{and} \quad F_1 = (1, 1, \ldots, 1)$$

be vectors in $\mathbb{R}^{10}$. Then in coordinates

$$Y - (b_1 F_1 + b_2 X) = \begin{pmatrix} y_1 - (b_1 + b_2 x_1) \\ \vdots \\ y_{10} - (b_1 + b_2 x_{10}) \end{pmatrix}.$$

It follows that

$$E(b) = ||Y - (b_1 F_1 + b_2 X)||.$$

The problem of making a least squares fit is to minimize E over all b_1 and b_2.

To solve the minimization problem, note that the vectors $b_1 F_1 + b_2 X$ form a two dimensional subspace $W = \text{span}\{F_1, X\} \subset \mathbb{R}^{10}$ (at least when X is not a scalar multiple of F_1, which is almost always). Minimizing E is identical to finding a vector $w_0 = b_1^0 F_1 + b_2^0 X \in W$ that is nearest to the vector $Y \in \mathbb{R}^{10}$. This is the least squares question that we solved in the Section 9.1.

We can use MATLAB to compute the values of b_1^0 and b_2^0 that give the best linear approximation to Y. If we set the matrix $A = (F_1|X)$, then Theorem 9.1.2 implies that the values of b_1^0 and b_2^0 are obtained using (9.1.4). In particular, type `e10_3_1` to call the vectors `X, Y, F1` into MATLAB, and then type

```
A = [F1 X];
b0 = inv(A'*A)*A'*Y
```

to obtain

```
b0(1) = -3.8597
b0(2) =  1.8845
```

Superimposing the line $y = -3.8597 + 1.8845x$ on the scatter plot in Figure 29 yields the plot in Figure 30. The total error is $E(b0) = 1.9634$ (obtained in MATLAB by typing `norm(Y-(b0(1)*F1+b0(2)*X))`. Compare this with the error $E(2, -4) = 2.0928$.

General Linear Regression We can summarize the previous discussion, as follows. Given n data points

$$(x_1, y_1), \ldots, (x_n, y_n);$$

form the vectors

$$X = (x_1, \ldots, x_n)^t \quad Y = (y_1, \ldots, y_n)^t \quad \text{and} \quad F_1 = (1, \ldots, 1)^t$$

in $\mathbb{R}^n$. Find constants b_1^0 and b_2^0 so that $b_1^0 F_1 + b_2^0 X$ is a vector in $W = \mathrm{span}\{F_1, X\} \subset \mathbb{R}^n$ that is nearest to Y. Let

$$A = (F_1|X)$$

be the $n \times 2$ matrix. This problem is solved by least squares in (9.1.4) as

$$\begin{pmatrix} b_1^0 \\ b_2^0 \end{pmatrix} = (A^t A)^{-1} A^t Y. \tag{9.2.2}$$

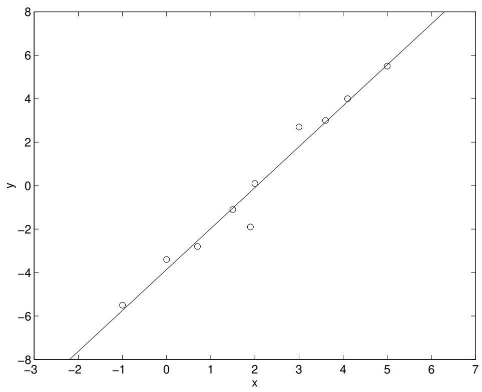

Figure 30: Scatter plot of data in (9.2.1*) with best linear approximation.

Least Squares Fit to a Quadratic Polynomial Suppose that we want to fit the data (x_i, y_i) to a quadratic polynomial

$$y = b_1 + b_2 x + b_3 x^2$$

by least squares methods. We want to find constants b_1^0, b_2^0, b_3^0 so that the error made is using the quadratic polynomial $y = b_1^0 + b_2^0 x + b_3^0 x^2$ is minimal among all possible choices of quadratic polynomials. The least squares error is

$$E(b) = ||Y - \left(b_1 F_1 + b_2 X + b_3 X^{(2)}\right)||$$

where

$$X^{(2)} = \left(x_1^2, \ldots, x_n^2\right)^t$$

and, as before, F_1 is the n vector with all components equal to 1.

We solve the minimization problem as before. In this case, the space of possible approximations to the data W is three dimensional; indeed, $W = \mathrm{span}\{F_1, X, X^{(2)}\}$. As in the case of fits to lines we try to find a point in W that is nearest to the vector $Y \in \mathbb{R}^n$. By (9.1.4), the answer is:

$$b = (A^tA)^{-1}A^tY,$$

where $A = (F_1|X|X^{(2)})$ is an $n \times 3$ matrix.

Suppose that we try to fit the data in (9.2.1*) with a quadratic polynomial rather than a linear one. Use MATLAB as follows

```
e9_3_1
A = [F1 X X.*X];
b = inv(A'*A)*A'*Y;
```

to obtain

```
b0(1) =   0.0443
b0(2) =   1.7054
b0(3) =  -3.8197
```

So the best parabolic fit to this data is $y = -3.8197 + 1.7054x + 0.0443x^2$. Note that the coefficient of x^2 is small suggesting that the data was well fit by a straight line. Note also that the error is $E(b0) = 1.9098$ which is only marginally smaller than the error for the best linear fit. For comparison, in Figure 31 we superimpose the equation for the quadratic fit onto Figure 30.

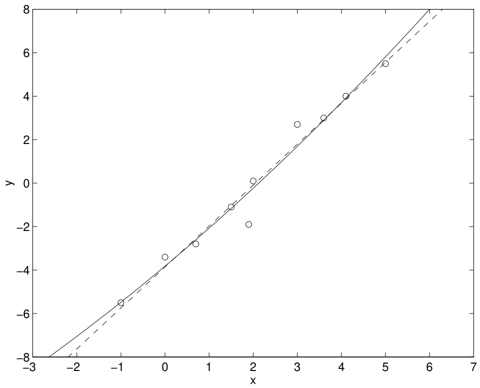

Figure 31: Scatter plot of data in (9.2.1*) with best linear and quadratic approximations. The best linear fit is plotted with a dashed line.

General Least Squares Fit The approximation to a quadratic polynomial shows that least squares fits can be made to any finite dimensional function space. More precisely, Let $\mathcal{C}$ be a finite dimensional space of functions and let

$$f_1(x), \ldots, f_m(x)$$

be a basis for $\mathcal{C}$. We have just considered two such spaces: $\mathcal{C} = \text{span}\{f_1(x) = 1, f_2(x) = x\}$ for linear regression and $\mathcal{C} = \text{span}\{f_1(x) = 1, f_2(x) = x, f_3(x) = x^2\}$ for least squares fit to a quadratic polynomial.

The general least squares fit of a data set

$$(x_1, y_1), \ldots, (x_n, y_n)$$

is the function $g_0(x) \in \mathcal{C}$ that is nearest to the data set in the following sense. Let

$$X = (x_1, \ldots, x_n)^t \quad \text{and} \quad Y = (y_1, \ldots, y_n)^t$$

be column vectors in $\mathbb{R}^n$. For any function $g(x)$ define the column vector

$$G = (g(x_1), \ldots, g(x_n))^t \in \mathbb{R}^n.$$

So G is the evaluation of $g(x)$ on the data set. Then the error

$$E(g) = ||Y - G||$$

is minimal for $g = g_0$.

More precisely, we think of the data Y as representing the (approximate) evaluation of a function on the x_i. Then we try to find a function $g_0 \in \mathcal{C}$ whose values on the x_i are as near as possible to the vector Y. This is just a least squares problem. Let $W \subset \mathbb{R}^n$ be the vector subspace spanned by the evaluations of function $g \in \mathcal{C}$ on the data points x_i, that is, the vectors G. The minimization problem is to find a vector in W that is nearest to Y. This can be solved in general using (9.1.4). That is, let A be the $n \times m$ matrix

$$A = (F_1|\cdots|F_m)$$

where $F_j \in \mathbb{R}^n$ is the column vector associated to the j^{th} basis element of $\mathcal{C}$, that is,

$$F_j = (f_j(x_1), \ldots, f_j(x_n))^t \in \mathbb{R}^n.$$

The minimizing function $g_0(x) \in \mathcal{C}$ is a linear combination of the basis functions $f_1(x), \ldots, f_n(x)$, that is,

$$g_0(x) = b_1 f_1(x) + \cdots + b_m f_m(x)$$

for scalars b_i. If we set

$$b = (b_1, \ldots, b_m) \in \mathbb{R}^m,$$

then least squares minimization states that

$$b = (A'A)^{-1}A'Y. \tag{9.2.3}$$

This equation can be solved easily in MATLAB. Enter the data as column n-vectors `X` and `Y`. Compute the column vectors `Fj = ` f_j`(X)` and then form the matrix `A = [F1 F2` $\cdots$ `Fm]`. Finally compute

```
b = inv(A'*A)*A'*Y
```

Least Squares Fit to a Sinusoidal Function We discuss a specific example of the general least squares formulation by considering the weather. It is reasonable to expect monthly data on the weather to vary periodically in time with a period of one year. In Table 3 we give average daily high and low temperatures for each month of the year for Paris and Rio de Janeiro. We attempt to fit this data with curves of the form:

$$g(T) = b_1 + b_2 \cos\left(\frac{2\pi}{12}T\right) + b_3 \sin\left(\frac{2\pi}{12}T\right),$$

where T is time measured in months and b_1, b_2, b_3 are scalars. These functions are 12 periodic, which seems appropriate for weather data, and form a three dimensional function space $\mathcal{C}$. Recall the trigonometric identity

$$a\cos(\omega t) + c\sin(\omega t) = d\sin(\omega(t - \varphi))$$

where

$$d = \sqrt{a^2 + c^2}.$$

Based on this identity we call $\mathcal{C}$ the space of *sinusoidal functions*. The number d is called the *amplitude* of the sinusoidal function $g(T)$.

Note that each data set consists of twelve entries — one for each month. Let $T = (1, 2, \ldots, 12)^t$ be the vector $X \in \mathbb{R}^{12}$ in the general presentation. Next let Y be the data in one of the data sets — say the high temperatures in Paris.

Now we turn to the vectors representing basis functions in $\mathcal{C}$. Let

```
F1=[1 1 1 1 1 1 1 1 1 1 1 1]'
```

be the vector associated with the basis function $f_1(T) = 1$. Let `F2` and `F3` be the column vectors associated to the basis functions

$$f_2(T) = \cos\left(\frac{2\pi}{12}T\right) \quad \text{and} \quad f_3(T) = \sin\left(\frac{2\pi}{12}T\right).$$

These vectors are computed by typing

```
F2 = cos(2*pi/12*T);
F3 = sin(2*pi/12*T);
```

By typing `temper`, we enter the temperatures and the vectors `T`, `F1`, `F2` and `F3` into MATLAB.

To find the best fit to the data by a sinusoidal function $g(T)$, we use (9.1.4). Let A be the 12×3 matrix

	Paris		Rio de Janeiro			Paris		Rio de Janeiro	
Month	High	Low	High	Low	Month	High	Low	High	Low
1	55	39	84	73	7	81	64	75	63
2	55	41	85	73	8	81	64	76	64
3	59	45	83	72	9	77	61	75	65
4	64	46	80	69	10	70	54	77	66
5	68	55	77	66	11	63	46	79	68
6	75	61	76	64	12	55	41	82	71

Table 3: Monthly Average of Daily High and Low Temperatures in Paris and Rio de Janeiro.

```
A = [F1 F2 F3];
```

The table data is entered in column vectors `ParisH` and `ParisL` for the high and low Paris temperatures and `RioH` and `RioL` for the high and low Rio de Janeiro temperatures. We can find the best least squares fit of the Paris high temperatures by a sinusoidal function $g_0(T)$ by typing

```
b = inv(A'*A)*A'*ParisH
```

obtaining

```
b(1) =  66.9167
b(2) =  -9.4745
b(3) =  -9.3688
```

The result is plotted in Figure 32 by typing

```
plot(T,ParisH,'o')
axis([0,13,0,100])
xlabel('time (months)')
ylabel('temperature (Fahrenheit)')
hold on
xx = linspace(0,13);
yy = b(1) + b(2)*cos(2*pi*xx/12) +
     b(3)*sin(2*pi*xx/12);
plot(xx,yy)
```

A similar exercise allows us to compute the best approximation to the Rio de Janeiro high temperatures obtaining

```
b(1) =  79.0833
b(2) =   3.0877
b(3) =   3.6487
```

The value of $b(1)$ is just the mean high temperature and not surprisingly that value is much higher in Rio than in Paris. There is yet more information contained in these approximations. For the high temperatures in Paris and Rio

$$d_P = 13.3244 \quad \text{and} \quad d_R = 4.7798.$$

The amplitude d measures the variation of the high temperature about its mean. It is much greater in Paris than in Rio, indicating that the difference in temperature between winter and summer is much greater in Paris than in Rio.

Least Squares Fit in MATLAB The general formula for a least squares fit of data (9.2.3) has been preprogrammed in MATLAB. After setting up the matrix A whose columns are the vectors F_j just type

```
b = A\Y
```

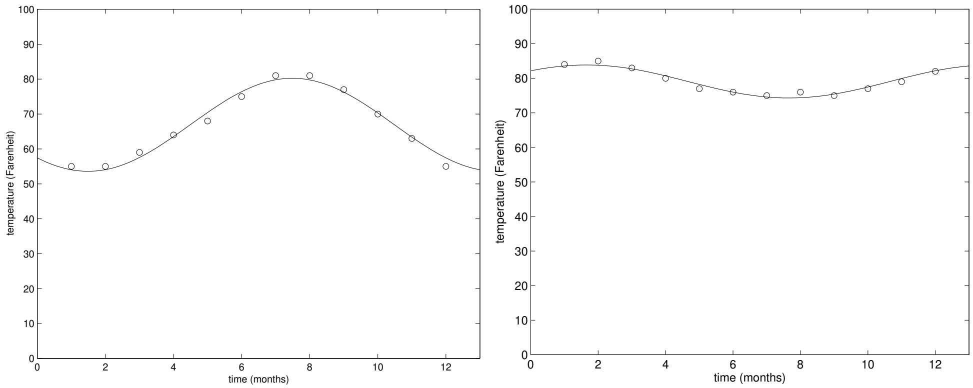

Figure 32: Monthly averages of daily high temperatures in Paris (left) and Rio de Janeiro (right) with best sinusoidal approximation.

This MATLAB command can be checked on the sinusoidal fit to the high temperature Rio de Janeiro data by typing

```
b = A\RioH
```

and obtaining

```
b =
   79.0833
    3.0877
    3.6487
```

Exercises

10 Orthogonality

In Section 10.1 we discuss orthonormal bases (bases in which each basis vector has unit length and any two basis vectors are perpendicular) and orthogonal matrices (matrices whose columns form an orthonormal basis). We will see that the computation of coordinates in an orthonormal basis is particularly straightforward. The Gram-Schmidt orthonormalization process for constructing orthonormal bases is presented in Section 10.2. We use orthogonality in Section 10.3 to study the eigenvalues and eigenvectors of symmetric matrices (the eigenvalues are real and the eigenvectors can be chosen to be orthonormal). The chapter ends with a discussion of the QR decomposition for finding orthonormal bases in Section 10.4. This decomposition leads to an algorithm that is numerically superior to Gram-Schmidt and is the one used in MATLAB.

10.1 Orthonormal Bases and Orthogonal Matrices

In Section 8.3 we discussed how to write the coordinates of a vector in a basis. We now show that finding coordinates of vectors in certain bases is a very simple task — these bases are called orthonormal bases.

Nonzero vectors $v_1, \ldots, v_k$ in $\mathbb{R}^n$ are *orthogonal* if the dot products

$$v_i \cdot v_j = 0$$

when $i \neq j$. The vectors are *orthonormal* if they are orthogonal and of unit length, that is,

$$v_i \cdot v_i = 1.$$

The standard example of a set of orthonormal vectors in $\mathbb{R}^n$ is the standard basis $e_1, \ldots, e_n$.

Lemma 10.1.1. *Nonzero orthogonal vectors are linearly independent.*

Proof Let $v_1, \ldots, v_k$ be a set of nonzero orthogonal vectors in $\mathbb{R}^n$ and suppose that

$$\alpha_1 v_1 + \cdots + \alpha_k v_k = 0.$$

To prove the lemma we must show that each $\alpha_j = 0$. Since $v_i \cdot v_j = 0$ for $i \neq j$,

$$\begin{aligned} \alpha_j v_j \cdot v_j &= \alpha_1 v_1 \cdot v_j + \cdots + \alpha_k v_k \cdot v_j \\ &= (\alpha_1 v_1 + \cdots + \alpha_k v_k) \cdot v_j = 0 \cdot v_j = 0. \end{aligned}$$

Since $v_j \cdot v_j = ||v_j||^2 > 0$, it follows that $\alpha_j = 0$. ■

Corollary 10.1.2. *A set of n nonzero orthogonal vectors in $\mathbb{R}^n$ is a basis.*

Proof Lemma 10.1.1 implies that the n vectors are linearly independent, and Chapter 5, Corollary 5.6.7 states that n linearly independent vectors in $\mathbb{R}^n$ form a basis. ■

Next we discuss how to find coordinates of a vector in an *orthonormal basis*, that is, a basis consisting of orthonormal vectors.

Theorem 10.1.3. *Let $V \subset \mathbb{R}^n$ be a subspace and let $\{v_1, \ldots, v_k\}$ be an orthonormal basis of V. Let $v \in V$ be a vector. Then*

$$v = \alpha_1 v_1 + \cdots + \alpha_k v_k.$$

where

$$\alpha_i = v \cdot v_i.$$

Proof Since $\{v_1, \ldots, v_k\}$ is a basis of V, we can write

$$v = \alpha_1 v_1 + \cdots + \alpha_k v_k$$

for some scalars α_j. It follows that

$$v \cdot v_j = (\alpha_1 v_1 + \cdots + \alpha_k v_k) \cdot v_j = \alpha_j,$$

as claimed. ■

An Example in $\mathbb{R}^3$ Let

$$\begin{aligned} v_1 &= \frac{1}{\sqrt{3}}(1, 1, 1), \\ v_2 &= \frac{1}{\sqrt{6}}(1, -2, 1), \\ v_3 &= \frac{1}{\sqrt{2}}(1, 0, -1). \end{aligned}$$

A short calculation verifies that these vectors have unit length and are pairwise orthogonal. Let $v = (1, 2, 3)$ be a vector and determine the coordinates of v in the basis $\mathcal{V} = \{v_1, v_2, v_3\}$. Theorem 10.1.3 states that these coordinates are:

$$[v]_{\mathcal{V}} = (v \cdot v_1, v \cdot v_2, v \cdot v_3) = (2\sqrt{3}, \frac{7}{\sqrt{6}}, -\sqrt{2}).$$

Matrices in Orthonormal Coordinates Next we discuss how to find the matrix associated with a linear map in an orthonormal basis. Let $L : \mathbb{R}^n \to \mathbb{R}^n$ be a linear map and let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be an orthonormal basis for $\mathbb{R}^n$. Then the matrix associated to L in the basis $\mathcal{V}$ can be calculated in terms of dot product. That matrix is:

$$[L]_{\mathcal{V}} = L(v_j) \cdot v_i. \qquad (10.1.1)$$

To verify (10.1.1), recall from Definition 8.3.3 that the $(i,j)^{th}$ entry of $[L]_{\mathcal{V}}$ is the i^{th} entry in the vector $[L(v_j)]_{\mathcal{V}}$ which is $L(v_j) \cdot v_i$ by Theorem 10.1.3.

An Example in $\mathbb{R}^2$ Let $\mathcal{V} = \{v_1, v_2\} \subset \mathbb{R}^2$ where

$$v_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad v_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

The set $\mathcal{V}$ is an orthonormal basis of $\mathbb{R}^2$. Using (10.1.1) we can find the matrix associated to the linear map

$$L_A(x) = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix} x$$

in the basis $\mathcal{V}$. That is, compute

$$[L]_{\mathcal{V}} = \begin{pmatrix} Av_1 \cdot v_1 & Av_2 \cdot v_1 \\ Av_1 \cdot v_2 & Av_2 \cdot v_2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 5 & -3 \\ 1 & 5 \end{pmatrix}.$$

Orthogonal Matrices

Definition 10.1.4. An $n \times n$ matrix Q is *orthogonal* if its columns form an orthonormal basis of $\mathbb{R}^n$.

The following lemma states elementary properties of orthogonal matrices. Particularly, an orthogonal matrix is invertible and it is straightforward to compute its inverse.

Lemma 10.1.5. *Let Q be an $n \times n$ matrix. Then*

(a) Q is orthogonal if and only if $Q^tQ = I_n$;

(b) Q is orthogonal if and only if $Q^{-1} = Q^t$;

(c) If Q_1, Q_2 are orthogonal matrices, then Q_1Q_2 is an orthogonal matrix.

Proof (a) Let $Q = (v_1|\cdots|v_n)$. Since Q is orthogonal, the v_j form an orthonormal basis. By direct computation note that $Q^tQ = \{(v_i \cdot v_j)\} = I_n$, since the v_j are orthonormal. Note that (b) is simply a restatement of (a).

(c) Now let Q_1, Q_2 be orthogonal. Then (a) implies

$$(Q_1Q_2)^t(Q_1Q_2) = Q_2^tQ_1^tQ_1Q_2 = Q_2^tI_nQ_2 = Q_2^tQ_2 = I_n,$$

thus proving (c). ■

Remarks Concerning MATLAB In the next section we prove that every vector subspace of $\mathbb{R}^n$ has an orthonormal basis (see Theorem 10.2.1), and we present a method for constructing such a basis (the Gram-Schmidt orthonormalization process). Here we note that certain commands in MATLAB produce bases for vector spaces. For those commands MATLAB always produces an orthonormal basis. For example, `null(A)` produces a basis for the null space of A. Take the 3×5 matrix

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 0 & 0 \end{pmatrix}. \qquad (10.1.2^*)$$

Since $\text{rank}(A) = 3$, it follows that the null space of A is two-dimensional. Typing `B = null(A)` in MATLAB produces

```
B =
   -0.4666         0
    0.6945    0.4313
   -0.2876   -0.3235
    0.3581   -0.6470
   -0.2984    0.5392
```

The columns of B form an orthonormal basis for the null space of A. This assertion can be checked by first typing

```
v1 = B(:, 1);
v2 = B(:, 2);
```

and then typing

```
norm(v1)
norm(v2)
dot(v1,v2)
A*v1
A*v2
```

yields answers $1, 1, 0,\ (0,0,0)^t, (0,0,0)^t$ (to within numerical accuracy). Recall that the MATLAB command `norm(v)` computes the norm of a vector `v`.

Exercises

10.2 Gram-Schmidt Orthonormalization Process

Suppose that $\mathcal{W} = \{w_1, \ldots, w_k\}$ is a basis for the subspace $V \subset \mathbb{R}^n$. There is a natural process by which the $\mathcal{W}$ basis can be transformed into an orthonormal basis $\mathcal{V}$ of V. This process proceeds inductively on the w_j; the orthonormal vectors $v_1, \ldots, v_k$ can be chosen so that

$$\mathrm{span}\{v_1, \ldots, v_j\} = \mathrm{span}\{w_1, \ldots, w_j\}$$

for each $j \leq k$. Moreover, the v_j are chosen using the theory of least squares that we have just discussed.

The Case $j = 2$ To gain a feeling for how the induction process works, we verify the case $j = 2$. Set

$$v_1 = \frac{1}{||w_1||} w_1; \tag{10.2.1}$$

so v_1 points in the same direction as w_1 and has unit length, that is, $v_1 \cdot v_1 = 1$. The normalization is shown in Figure 33.

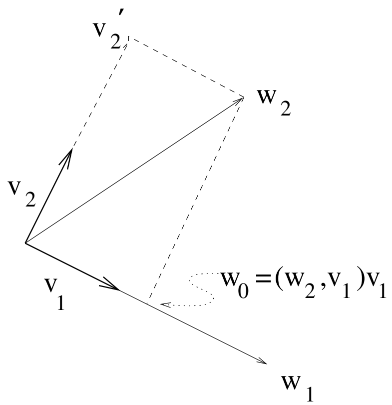

Figure 33: Planar illustration of Gram-Schmidt orthonormalization.

Next, we find a unit length vector v_2' in the plane spanned by w_1 and w_2 that is perpendicular to v_1. Let w_0 be the vector on the line generated by v_1 that is nearest to w_2. It follows from (9.1.3) that

$$w_0 = \frac{w_2 \cdot v_1}{||v_1||^2} v_1 = (w_2 \cdot v_1) v_1.$$

The vector w_0 is shown on Figure 33 and, as Lemma 9.1.1 states, the vector $v_2' = w_2 - w_0$ is perpendicular to v_1. That is,

$$v_2' = w_2 - (w_2 \cdot v_1) v_1 \tag{10.2.2}$$

is orthogonal to v_1.

Finally, set

$$v_2 = \frac{1}{||v_2'||} v_2' \tag{10.2.3}$$

so that v_2 has unit length. Since v_2 and v_2' point in the same direction, v_1 and v_2 are orthogonal. Note also that v_1 and v_2 are linear combinations of w_1 and w_2. Since v_1 and v_2 are orthogonal, they are linearly independent. It follows that

$$\mathrm{span}\{v_1, v_2\} = \mathrm{span}\{w_1, w_2\}.$$

In summary: computing v_1 and v_2 using (10.2.1), (10.2.2) and (10.2.3) yields an orthonormal basis for the plane spanned by w_1 and w_2.

The General Case

Theorem 10.2.1. *(Gram-Schmidt Orthonormalization) Let $w_1, \ldots, w_k$ be a basis for the subspace $W \subset \mathbb{R}^n$. Define v_1 as in (10.2.1) and then define inductively*

$$v_{j+1}' = w_{j+1} - [(w_{j+1} \cdot v_1) v_1 + \cdots + (w_{j+1} \cdot v_j) v_j] \tag{10.2.4}$$

$$v_{j+1} = \frac{1}{||v_{j+1}'||} v_{j+1}'. \tag{10.2.5}$$

Then $\operatorname{span}\{v_1,\ldots,v_j\} = \operatorname{span}\{w_1,\ldots,w_j\}$ *and* $v_1,\ldots,v_k$ *is an orthonormal basis of* W.

Proof We assume that we have constructed orthonormal vectors $v_1,\ldots,v_j$ such that

$$\operatorname{span}\{v_1,\ldots,v_j\} = \operatorname{span}\{w_1,\ldots,w_j\}.$$

Our purpose is to find a unit vector v_{j+1} that is orthogonal to each v_i and that satisfies

$$\operatorname{span}\{v_1,\ldots,v_{j+1}\} = \operatorname{span}\{w_1,\ldots,w_{j+1}\}.$$

We construct v_{j+1} in two steps. First we find a vector v'_{j+1} that is orthogonal to each of the v_i using least squares. Let w_0 be the vector in $\operatorname{span}\{v_1,\ldots,v_j\}$ that is nearest to w_{j+1}. Theorem 9.1.2 tells us how to make this construction. Let A be the matrix whose columns are $v_1,\ldots,v_j$. Then (9.1.4) states that the coordinates of w_0 in the v_i basis is given by $(A^tA)^{-1}A^tw_{j+1}$. But since the v_i's are orthonormal, the matrix A^tA is just I_k. Hence

$$w_0 = (w_{j+1}\cdot v_1)v_1 + \cdots + (w_{j+1}\cdot v_j)v_j.$$

Note that $v'_{j+1} = w_{j+1} - w_0$ is the vector defined in (10.2.4). We claim that $v'_{j+1} = w_{j+1} - w_0$ is orthogonal to v_k for $k \leq j$ and hence to every vector in $\operatorname{span}\{v_1,\ldots,v_j\}$. Just calculate

$$v'_{j+1}\cdot v_k = w_{j+1}\cdot v_k - w_0\cdot v_k = w_{j+1}\cdot v_k - w_{j+1}\cdot v_k = 0.$$

Define v_{j+1} as in (10.2.5). It follows that $v_1,\ldots,v_{j+1}$ are orthonormal and that each vector is a linear combination of $w_1,\ldots,w_{j+1}$. ■

An Example of Orthonormalization Let $W \subset \mathbb{R}^4$ be the subspace spanned by the vectors

$$w_1 = (1,0,-1,0), \quad w_2 = (2,-1,0,1), \quad w_3 = (0,0,-2,1). \tag{10.2.6}$$

We find an orthonormal basis for W using Gram-Schmidt orthonormalization.

Step 1: Set

$$v_1 = \frac{1}{||w_1||}w_1 = \frac{1}{\sqrt{2}}(1,0,-1,0).$$

Step 2: Following the Gram-Schmidt process, use (10.2.4) to define

$$\begin{aligned} v'_2 &= w_2 - (w_2\cdot v_1)v_1 \\ &= (2,-1,0,1) - \sqrt{2}\frac{1}{\sqrt{2}}(1,0,-1,0) \\ &= (1,-1,1,1). \end{aligned}$$

Normalization using (10.2.5) yields

$$v_2 = \frac{1}{||v'_2||}v'_2 = \frac{1}{2}(1,-1,1,1).$$

Step 3: Using (10.2.4) set

$$\begin{aligned} v'_3 &= w_3 - (w_3\cdot v_1)v_1 - (w_3\cdot v_2)v_2 \\ &= (0,0,-2,1) - \sqrt{2}\frac{1}{\sqrt{2}}(1,0,-1,0) - \\ &\quad \left(-\frac{1}{2}\right)\frac{1}{2}(1,-1,1,1) \\ &= \frac{1}{4}(-3,-1,-3,5). \end{aligned}$$

Normalization using (10.2.5) yields

$$v_3 = \frac{1}{||v'_3||}v'_3 = \frac{4}{\sqrt{44}}(-3,-1,-3,5).$$

Hence we have constructed an orthonormal basis

$\{v_1, v_2, v_3\}$ for W, namely

$$
\begin{aligned}
v_1 &= \frac{1}{\sqrt{2}}(1, 0, -1, 0) \\
&\approx (0.7071, 0, -0.7071, 0) \\
v_2 &= \frac{1}{2}(1, -1, 1, 1) \\
&= (0.5, -0.5, 0.5, 0.5) \\
v_3 &= \frac{4}{\sqrt{44}}(-3, -1, -3, 5) \\
&\approx (-0.4523, -0.1508, -0.4523, 0.7538)
\end{aligned}
$$

(10.2.7)

Exercises

10.3 The Spectral Theory of Symmetric Matrices

Eigenvalues and eigenvectors of symmetric matrices have remarkable properties that can be summarized in three theorems.

Theorem 10.3.1. *Let A be a symmetric matrix. Then every eigenvalue of A is real.*

Theorem 10.3.2. *Let A be an $n \times n$ symmetric matrix. Then there is an orthonormal basis of $\mathbb{R}^n$ consisting of eigenvectors of A.*

Theorem 10.3.3. *For each $n \times n$ symmetric matrix A, there exists an orthogonal matrix P such that P^tAP is a diagonal matrix.*

The proof of Theorem 10.3.1 uses the *Hermitian inner product* — a generalization of dot product to complex vectors.

Hermitian Inner Products Let $v, w \in \mathbb{C}^n$ be two complex n-vectors. Define

$$\langle v, w\rangle = v_1\overline{w}_1 + \cdots + v_n\overline{w}_n.$$

Note that the coordinates w_i of the second vector enter this formula with a complex conjugate. However, if v and w are real vectors, then

$$\langle v, w\rangle = v \cdot w.$$

An alternative notation for the Hermitian inner product is given by matrix multiplication. Suppose that v and w are column n-vectors. Then

$$\langle v, w\rangle = v^t\overline{w}.$$

The properties of the Hermitian inner product are similar to those of dot product. We note three. Let $c \in \mathbb{C}$ be a complex scalar. Then

$$\begin{aligned} \langle v, v\rangle &= ||v||^2 \geq 0 \\ \langle cv, w\rangle &= c\langle v, w\rangle \\ \langle v, cw\rangle &= \overline{c}\langle v, w\rangle \end{aligned}$$

Note the complex conjugation of the complex scalar c in the previous formula.

Let C be a complex $n \times n$ matrix. Then the most important observation concerning Hermitian inner products that we shall use is:

$$\langle Cv, w\rangle = \langle v, \overline{C}^t w\rangle. \tag{10.3.1}$$

This fact is verified by calculating

$$\langle Cv, w\rangle = (Cv)^t\overline{w} = (v^tC^t)\overline{w} = v^t(C^t\overline{w}) = v^t(\overline{\overline{C}^t w}) = \langle v, \overline{C}^t w\rangle.$$

So if A is a $n \times n$ real symmetric matrix, then

$$\langle Av, w\rangle = \langle v, Aw\rangle, \tag{10.3.2}$$

since $\overline{A}^t = A^t = A$.

Proof of Theorem 10.3.1 Let λ be an eigenvalue of A and let v be the associated eigenvector. Since $Av = \lambda v$ we can use (10.3.2) to compute

$$\lambda\langle v, v\rangle = \langle Av, v\rangle = \langle v, Av\rangle = \overline{\lambda}\langle v, v\rangle.$$

Since $\langle v, v\rangle = ||v||^2 > 0$, it follows that $\lambda = \overline{\lambda}$ and λ is real. ■

Proof of Theorem 10.3.2 Let A be a real symmetric $n \times n$ matrix. We show that there is an orthonormal basis of $\mathbb{R}^n$ consisting of eigenvectors of A. The proof follows directly from Corollary 10.1.2 if the eigenvalues are distinct.

If some of the eigenvalues are multiple, the proof is more complicated and uses Gram-Schmidt orthonormalization.

The proof proceeds inductively on n. The theorem is trivially valid for $n = 1$; so we assume that it is valid for $n - 1$.

Theorem 7.2.4 of Chapter 7 implies that A has an eigenvalue λ_1 and Theorem 10.3.1 states that this eigenvalue is real. Let v_1 be a unit length eigenvector corresponding to the eigenvalue λ_1. Extend v_1 to an orthonormal basis $v_1, w_2, \ldots, w_n$ of $\mathbb{R}^n$ and let $P = (v_1|w_2|\cdots|w_n)$ be the matrix whose columns are the vectors in this orthonormal basis. Orthonormality and direct multiplication implies that

$$P^tP = I_n. \tag{10.3.3}$$

Therefore P is invertible; indeed $P^{-1} = P^t$. Next, let

$$B = P^{-1}AP.$$

By direct computation

$$Be_1 = P^{-1}APe_1 = P^{-1}Av_1 = \lambda_1 P^{-1}v_1 = \lambda_1 e_1.$$

It follows that that B has the form

$$B = \begin{pmatrix} \lambda_1 & * \\ 0 & C \end{pmatrix}$$

where C is an $(n-1)\times(n-1)$ matrix. Since $P^{-1} = P^t$, it follows that B is a symmetric matrix; to verify this point compute

$$B^t = (P^tAP)^t = P^tA^t(P^t)^t = P^tAP = B.$$

It follows that

$$B = \begin{pmatrix} \lambda_1 & 0 \\ 0 & C \end{pmatrix}$$

where C is a symmetric matrix. By induction we can use the Gram-Schmidt orthonormalization process to choose an orthonormal basis $z_2, \ldots, z_n$ in $\{0\}\times\mathbb{R}^{n-1}$ consisting of eigenvectors of C. It follows that $e_1, z_2, \ldots, z_n$ is an orthonormal basis for $\mathbb{R}^n$ consisting of eigenvectors of B.

Finally, let $v_j = P^{-1}z_j$ for $j = 2, \ldots, n$. Since $v_1 = P^{-1}e_1$, it follows that $v_1, v_2, \ldots, v_n$ is a basis of $\mathbb{R}^n$ consisting of eigenvectors of A. We need only show that the v_j form an orthonormal basis of $\mathbb{R}^n$. This is done using (10.3.2). For notational convenience let $z_1 = e_1$ and compute

$$\begin{aligned} \langle v_i, v_j\rangle &= \langle P^{-1}z_i, P^{-1}z_j\rangle = \langle P^tz_i, P^tz_j\rangle \\ &= \langle z_i, PP^tz_j\rangle = \langle z_i, z_j\rangle, \end{aligned}$$

since $PP^t = I_n$. Thus the vectors v_j form an orthonormal basis since the vectors z_j form an orthonormal basis. ■

Proof of Theorem 10.3.3 As a consequence of Theorem 10.3.2, let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be an orthonormal basis for $\mathbb{R}^n$ consisting of eigenvectors of A. Indeed, suppose

$$Av_j = \lambda_j v_j$$

where $\lambda_j \in \mathbb{R}$. Note that

$$Av_j \cdot v_i = \begin{cases} \lambda_j & i = j \\ 0 & i \neq j \end{cases}$$

It follows from (10.1.1) that

$$[A]_{\mathcal{V}} = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$$

is a diagonal matrix. So every symmetric matrix A is similar by an orthogonal matrix P to a diagonal matrix where P is the matrix whose columns are the eigenvectors of A; namely, $P = [v_1|\cdots|v_n]$. ■

Exercises

10.4 *QR Decompositions

In this section we describe an alternative approach to Gram-Schmidt orthonormalization for constructing an orthonormal basis of a subspace $W \subset \mathbb{R}^n$. This method is called the QR decomposition and is numerically superior to Gram-Schmidt. Indeed, the QR decomposition is the method used by MATLAB to compute orthonormal bases. To discuss this decomposition we need to introduce a new type of matrices, the orthogonal matrices.

Reflections Across Hyperplanes: Householder Matrices

Useful examples of orthogonal matrices are reflections across hyperplanes. An $n-1$ dimensional subspace of $\mathbb{R}^n$ is called a *hyperplane*. Let V be a hyperplane and let u be a nonzero vector normal to V. Then a *reflection* across V is a linear map $H : \mathbb{R}^n \to \mathbb{R}^n$ such that

(a) $Hv = v$ for all $v \in V$.

(b) $Hu = -u$.

We claim that the matrix of a reflection across a hyperplane is orthogonal and there is a simple formula for that matrix.

Definition 10.4.1. A *Householder matrix* is an $n \times n$ matrix of the form

$$H = I_n - \frac{2}{u^t u} u u^t \tag{10.4.1}$$

where $u \in \mathbb{R}^n$ is a nonzero vector. .

This definition makes sense since $u^t u = ||u||^2$ is a number while the product uu^t is an $n \times n$ matrix.

Lemma 10.4.2. *Let $u \in \mathbb{R}^n$ be a nonzero vector and let V be the hyperplane orthogonal to u. Then the Householder matrix H is a reflection across V and is orthogonal.*

Proof By definition every vector $v \in V$ satisfies $u^t v = u \cdot v = 0$. Therefore,

$$Hv = v - \frac{2}{u^t u} u u^t v = v,$$

and

$$Hu = u - \frac{2}{u^t u} u u^t u = u - 2u = -u.$$

Hence H is a reflection across the hyperplane V. It also follows that $H^2 = I_n$ since $H^2 v = H(Hv) = Hv = v$ for all $v \in V$ and $H^2 u = H(-u) = u$. So H^2 acts like the identity on a basis of $\mathbb{R}^n$ and $H^2 = I_n$.

To show that H is orthogonal, we first calculate

$$H^t = I_n^t - \frac{2}{u^t u}(uu^t)^t = I_n - \frac{2}{u^t u} u u^t = H.$$

Therefore $I_n = HH = HH^t$ and $H^t = H^{-1}$. Now apply Lemma 10.1.5(b). ■

QR Decompositions The Gram-Schmidt process is not used in practice to find orthonormal bases as there are other techniques available that are preferable for orthogonalization on a computer. One such procedure for the construction of an orthonormal basis is based on QR decompositions using *Householder transformations*. This method is the one implemented in MATLAB .

An $n \times k$ matrix $R = \{r_{ij}\}$ is *upper triangular* if $r_{ij} = 0$ whenever $i > j$.

Definition 10.4.3. An $n \times k$ matrix A has a *QR decomposition* if

$$A = QR. \tag{10.4.2}$$

where Q is an $n \times n$ orthogonal matrix and R is an $n \times k$ upper triangular matrix R.

QR decompositions can be used to find orthonormal bases as follows. Suppose that $\mathcal{W} = \{w_1, \ldots, w_k\}$ is a basis for the subspace $W \subset \mathbb{R}^n$. Then define the $n \times k$ matrix A which has the w_j as columns, that is

$$A = (w_1^t|\cdots|w_k^t).$$

Suppose that $A = QR$ is a QR decomposition. Since Q is orthogonal, the columns of Q are orthonormal. So write

$$Q = (v_1^t|\cdots|v_n^t).$$

On taking transposes we arrive at the equation $A^t = R^tQ^t$:

$$\begin{pmatrix} w_1 \\ \vdots \\ w_k \end{pmatrix} = \begin{pmatrix} r_{11} & 0 & \cdots & 0 & \cdots & 0 \\ r_{12} & r_{22} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ r_{1k} & r_{2k} & \cdots & r_{kk} & \cdots & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}.$$

By equating rows in this matrix equation we arrive at the system

$$\begin{aligned} w_1 &= r_{11}v_1 \\ w_2 &= r_{12}v_1 + r_{22}v_2 \\ &\vdots \\ w_k &= r_{1k}v_1 + r_{2k}v_2 + \cdots + r_{kk}v_k. \end{aligned} \tag{10.4.3}$$

It now follows that the $W = \mathrm{span}\{v_1, \ldots, v_k\}$ and that $\{v_1, \ldots, v_k\}$ is an orthonormal basis for W. We have proved:

Proposition 10.4.4. *Suppose that there exist an orthogonal $n\times n$ matrix Q and an upper triangular $n\times k$ matrix R such that the $n \times k$ matrix A has a QR decomposition*

$$A = QR.$$

Then the first k columns $v_1, \ldots, v_k$ of the matrix Q form an orthonormal basis of the subspace $W = \mathrm{span}\{w_1, \ldots, w_k\}$, where the w_j are the columns of A. Moreover, $r_{ij} = v_i \cdot w_j$ is the coordinate of w_j in the orthonormal basis.

Conversely, we can also write down a QR decomposition for a matrix A, if we have computed an orthonormal basis for the columns of A. Indeed, using the Gram-Schmidt process, Theorem 10.2.1, we have shown that QR decompositions always exist. In the remainder of this section we discuss a different way for finding QR decompositions using Householder matrices.

Construction of a QR Decomposition Using Householder Matrices The QR decomposition by Householder transformations is based on the following observation :

Proposition 10.4.5. *Let $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$ be nonzero and let*

$$r = \sqrt{z_j^2 + \cdots + z_n^2}.$$

Define $u = (u_1, \ldots, u_n) \in \mathbb{R}^n$ by

$$\begin{pmatrix} u_1 \\ \vdots \\ u_{j-1} \\ u_j \\ u_{j+1} \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ z_j - r \\ z_{j+1} \\ \vdots \\ z_n \end{pmatrix}.$$

Then

$$2u^tz = u^tu$$

and

$$Hz = \begin{pmatrix} z_1 \\ \vdots \\ z_{j-1} \\ r \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{10.4.4}$$

holds for the Householder matrix $H = I_n - \dfrac{2}{u^tu}uu^t$.

Proof Begin by computing

$$\begin{aligned} u^t z &= u_j z_j + z_{j+1}^2 + \cdots + z_n^2 \\ &= z_j^2 - r z_j + z_{j+1}^2 + \cdots + z_n^2 \\ &= -r z_j + r^2. \end{aligned}$$

Next, compute

$$\begin{aligned} u^t u &= (z_j - r)(z_j - r) + z_{j+1}^2 + \cdots + z_n^2 \\ &= z_j^2 - 2r z_j + r^2 + z_{j+1}^2 + \cdots + z_n^2 \\ &= 2(-r z_j + r^2). \end{aligned}$$

Hence $2u^t z = u^t u$, as claimed.

Note that $z - u$ is the vector on the right hand side of (10.4.4). So, compute

$$Hz = \left(I_n - \frac{2}{u^t u} u u^t \right) z = z - \frac{2u^t z}{u^t u} u = z - u$$

to see that (10.4.4) is valid. ∎

An inspection of the proof of Proposition 10.4.5 shows that we could have chosen

$$u_j = z_j + r$$

instead of $u_j = z_j - r$. Therefore, the choice of H is not unique.

Proposition 10.4.5 allows us to determine inductively a QR decomposition of the matrix

$$A = (w_1^0 | \cdots | w_k^0),$$

where each $w_j^0 \in \mathbb{R}^n$. So, A is an $n \times k$ matrix and $k \leq n$.

First, set $z = w_1^0$ and use Proposition 10.4.5 to construct the Householder matrix H_1 such that

$$H_1 w_1^0 = \begin{pmatrix} r_{11} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \equiv r_1.$$

Then the matrix $A_1 = H_1 A$ can be written as

$$A_1 = (r_1 | w_2^1 | \cdots | w_k^1),$$

where $w_j^1 = H_1 w_j^0$ for $j = 2, \ldots, k$.

Second, set $z = w_2^1$ in Proposition 10.4.5 and construct the Householder matrix H_2 such that

$$H_2 w_2^1 = \begin{pmatrix} r_{12} \\ r_{22} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \equiv r_2.$$

Then the matrix $A_2 = H_2 A_1 = H_2 H_1 A$ can be written as

$$A_2 = (r_1 | r_2 | w_3^2 | \cdots | w_k^2)$$

where $w_j^2 = H_2 w_j^1$ for $j = 3, \ldots, k$. Observe that the 1^{st} column r_1 is not affected by the matrix multiplication, since H_2 leaves the first component of a vector unchanged.

Proceeding inductively, in the i^{th} step, set $z = w_i^{i-1}$ and use Proposition 10.4.5 to construct the Householder matrix H_i such that:

$$H_i w_i^{i-1} = \begin{pmatrix} r_{1i} \\ \vdots \\ r_{ii} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \equiv r_i$$

and the matrix $A_i = H_i A_{i-1} = H_i \cdots H_1 A$ can be written as

$$A_i = (r_1 | \cdots | r_i | w_{i+1}^i | \cdots | w_k^i),$$

where $w_i^2 = H_i w_j^{i-1}$ for $j = i+1, \ldots, k$.

After k steps we arrive at

$$H_k \cdots H_1 A = R,$$

where $R = (r_1|\cdots|r_k)$ is an upper triangular $n \times k$ matrix. Since the Householder matrices $H_1, \ldots, H_k$ are orthogonal, it follows from Lemma 10.1.5(c) that the $Q^t = H_k \cdots H_1$ is orthogonal. Thus, $A = QR$ is a QR decomposition of A.

Orthonormalization with MATLAB Given a set $w_1, \ldots, w_k$ of linearly independent vectors in $\mathbb{R}^n$ the MATLAB command `qr` allows us to compute an orthonormal basis of the spanning set of these vectors. As mentioned earlier, the underlying technique MATLAB uses for the computation of the QR decomposition is based on Householder transformations.

The syntax of the QR decomposition in MATLAB is quite simple. For example, let $w_1 = (1, 0, -1, 0)$, $w_2 = (2, -1, 0, 1)$ and $w_3 = (0, 0, -2, 1)$ be the three vectors in (10.2.6). In Section 5.5 we computed an orthonormal basis for the subspace of $\mathbb{R}^4$ spanned by w_1, w_2, w_3. Here we use the MATLAB command `qr` to find an orthonormal basis for this subspace. Let A be the matrix having the vectors w_1^t, w_2^t and w_3^t as columns. So, A is:

```
A = [1 2 0; 0 -1 0; -1 0 -2; 0 1 1]
```

The command

```
[Q R] = qr(A,0)
```

leads to the answer

```
Q =
   -0.7071    0.5000   -0.4523
         0   -0.5000   -0.1508
    0.7071    0.5000   -0.4523
         0    0.5000    0.7538

R =
   -1.4142   -1.4142   -1.4142
         0    2.0000   -0.5000
         0         0    1.6583
```

A comparison with (10.2.7) shows that the columns of the matrix `Q` are the elements in the orthonormal basis. The only difference is that the sign of the first vector is opposite. However, this is not surprising since we know that there is some freedom in the choice of Householder matrices, as remarked after Proposition 10.4.5.

In addition, the command `qr` produces the matrix `R` whose entries r_{ij} are the coordinates of the vectors w_j in the new orthonormal basis as in (10.4.3). For instance, the second column of `R` tells us that

$$w_2 = r_{12}v_1 + r_{22}v_2 + r_{32}v_3 = -1.4142v_1 + 2.0000v_2.$$

Exercises

11 *Matrix Normal Forms

In this chapter we generalize to $n \times n$ matrices the theory of matrix normal forms presented in Chapter 6 for 2×2 matrices. In this theory we ask: What is the simplest form that a matrix can have up to *similarity*. After first presenting several preliminary results, the theory culminates in the Jordan normal form theorem, Theorem 11.3.2.

The first of the matrix normal form results — every matrix with n distinct real eigenvalues can be diagonalized — is presented in Section 7.3. The basic idea is that when a matrix has n distinct real eigenvalues, then it has n linearly independent eigenvectors. In Section 11.1 we discuss matrix normal forms when the matrix has n distinct eigenvalues some of which are complex. When an $n \times n$ matrix has fewer than n linearly independent eigenvectors, it must have multiple eigenvalues and generalized eigenvectors. This topic is discussed in Section 11.2. The Jordan normal form theorem is introduced in Section 11.3 and describes similarity of matrices when the matrix has fewer than n independent eigenvectors. The proof is given in Appendix 11.5.

We introduced Markov matrices in Section 4.8. One of the theorems discussed there has a proof that relies on the Jordan normal form theorem, and we prove this theorem in Appendix 11.4.

11.1 Simple Complex Eigenvalues

Theorem 7.3.1 states that a matrix A with real unequal eigenvalues may be diagonalized. It follows that in an appropriately chosen basis (the basis of eigenvectors), matrix multiplication by A acts as multiplication by these real eigenvalues. Moreover, geometrically, multiplication by A stretches or contracts vectors in eigendirections (depending on whether the eigenvalue is greater than or less than 1 in absolute value).

The purpose of this section is to show that a similar kind of diagonalization is possible when the matrix has distinct complex eigenvalues. See Theorem 11.1.1 and Theorem 11.1.2. We show that multiplication by a matrix with complex eigenvalues corresponds to multiplication by complex numbers. We also show that multiplication by complex eigenvalues correspond geometrically to rotations as well as expansions and contractions.

The Geometry of Complex Eigenvalues: Rotations and Dilatations Real 2×2 matrices are the smallest real matrices where complex eigenvalues can possibly occur. In Chapter 6, Theorem 6.3.4(b) we discussed the classification of such matrices up to similarity. Recall that if the eigenvalues of a 2×2 matrix A are $\sigma \pm i\tau$, then A is similar to the matrix

$$\begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix}. \tag{11.1.1}$$

Moreover, the basis in which A has the form (11.1.1) is found as follows. Let $v = w_1 + iw_2$ be the eigenvector of A corresponding to the eigenvalue $\sigma - i\tau$. Then $\{w_1, w_2\}$ is the desired basis.

Geometrically, multiplication of vectors in $\mathbb{R}^2$ by (11.1.1) is the same as a rotation followed by a dilatation. More specifically, let $r = \sqrt{\sigma^2+\tau^2}$. So the point (σ,τ) lies on the circle of radius r about the origin, and there is an angle θ such that $(\sigma,\tau) = (r\cos\theta, r\sin\theta)$. Now we can rewrite (11.1.1) as

$$\begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix} = r\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = rR_\theta,$$

where R_θ is rotation counterclockwise through angle θ. From this discussion we see that geometrically complex eigenvalues are associated with rotations followed either by stretching ($r > 1$) or contracting ($r < 1$).

As an example, consider the matrix

$$A = \begin{pmatrix} 2 & 1 \\ -2 & 0 \end{pmatrix}. \tag{11.1.2}$$

The characteristic polynomial of A is $p_A(\lambda) = \lambda^2 - 2\lambda + 2$. Thus the eigenvalues of A are $1 \pm i$, and $\sigma = 1$ and $\tau = 1$ for this example. An eigenvector associated to the eigenvalue $1-i$ is $v = (1, -1-i)^t = (1,-1)^t + i(0,-1)^t$. Therefore,

$$B = S^{-1}AS = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad \text{where} \quad S = \begin{pmatrix} 1 & 0 \\ -1 & -1 \end{pmatrix},$$

as can be checked by direct calculation. Moreover, we can rewrite

$$B = \sqrt{2}\begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} = \sqrt{2}R_{\frac{\pi}{4}}.$$

So, in an appropriately chosen coordinate system, multiplication by A rotates vectors counterclockwise by 45° and then expands the result by a factor of $\sqrt{2}$. See Exercise **??**.

The Algebra of Complex Eigenvalues: Complex Multiplication We have shown that the normal form (11.1.1) can be interpreted geometrically as a rotation followed by a dilatation. There is a second algebraic interpretation of

(11.1.1), and this interpretation is based on multiplication by complex numbers.

Let $\lambda = \sigma + i\tau$ be a complex number and consider the matrix associated with complex multiplication, that is, the linear mapping

$$z \mapsto \lambda z \tag{11.1.3}$$

on the complex plane. By identifying real and imaginary parts, we can rewrite (11.1.3) as a real 2×2 matrix in the following way. Let $z = x + iy$. Then

$$\lambda z = (\sigma + i\tau)(x + iy) = (\sigma x - \tau y) + i(\tau x + \sigma y).$$

Now identify z with the vector (x, y); that is, the vector whose first component is the real part of z and whose second component is the imaginary part. Using this identification the complex number λz is identified with the vector $(\sigma x - \tau y, \tau x + \sigma y)$. So, in real coordinates and in matrix form, (11.1.3) becomes

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \sigma x - \tau y \\ \tau x + \sigma y \end{pmatrix} = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

That is, the matrix corresponding to multiplication of $z = x + iy$ by the complex number $\lambda = \sigma + i\tau$ is the one that multiplies the vector $(x, y)^t$ by the normal form matrix (11.1.1).

Direct Agreement Between the Two Interpretations of (11.1.1) We have shown that matrix multiplication by (11.1.1) may be thought of either algebraically as multiplication by a complex number (an eigenvalue) or geometrically as a rotation followed by a dilatation. We now show how to go directly from the algebraic interpretation to the geometric interpretation.

Euler's formula (Chapter 6, (6.2.5)) states that

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

for any real number θ. It follows that we can write a complex number $\lambda = \sigma + i\tau$ in polar form as

$$\lambda = re^{i\theta}$$

where $r^2 = \lambda\overline{\lambda} = \sigma^2 + \tau^2$, $\sigma = r\cos\theta$, and $\tau = r\sin\theta$.

Now consider multiplication by λ in polar form. Write $z = se^{i\varphi}$ in polar form, and compute

$$\lambda z = re^{i\theta}se^{i\varphi} = rse^{i(\varphi+\theta)}.$$

It follows from polar form that multiplication of z by $\lambda = re^{i\theta}$ rotates z through an angle θ and dilates the result by the factor r. Thus Euler's formula directly relates the geometry of rotations and dilatations with the algebra of multiplication by a complex number.

Normal Form Matrices with Distinct Complex Eigenvalues In the first parts of this section we have discussed a geometric and an algebraic approach to matrix multiplication by 2×2 matrices with complex eigenvalues. We now turn our attention to classifying $n \times n$ matrices that have distinct eigenvalues, whether these eigenvalues are real or complex. We will see that there are two ways to frame this classification — one algebraic (using complex numbers) and one geometric (using rotations and dilatations).

Algebraic Normal Forms: The Complex Case Let A be an $n \times n$ matrix with real entries and n distinct eigenvalues $\lambda_1, \ldots, \lambda_n$. Let v_j be an eigenvector associated with the eigenvalue λ_j. By methods that are entirely analogous to those in Section 7.3 we can diagonalize the matrix A over the complex numbers. The resulting theorem is analogous to Theorem 7.3.1.

More precisely, the $n \times n$ matrix A is *complex diagonal-*

izable if there is a complex $n \times n$ matrix T such that

$$T^{-1}AT = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}.$$

Theorem 11.1.1. *Let A be an $n \times n$ matrix with n distinct eigenvalues. Then A is complex diagonalizable.*

The proof of Theorem 11.1.1 follows from a theoretical development virtually word for word the same as that used to prove Theorem 7.3.1 in Section 7.3. Beginning from the theory that we have developed so far, the difficulty in proving this theorem lies in the need to base the theory of linear algebra on complex scalars rather than real scalars. We will not pursue that development here.

As in Theorem 7.3.1, the proof of Theorem 11.1.1 shows that the complex matrix T is the matrix whose columns are the eigenvectors v_j of A; that is,

$$T = (v_1|\cdots|v_n).$$

Finally, we mention that the computation of inverse matrices with complex entries is the same as that for matrices with real entries. That is, row reduction of the $n \times 2n$ matrix $(T|I_n)$ leads, when T is invertible, to the matrix $(I_n|T^{-1})$.

Two Examples As a first example, consider the normal form 2×2 matrix (11.1.1) that has eigenvalues λ and $\overline{\lambda}$, where $\lambda = \sigma + i\tau$. Let

$$B = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} \lambda & 0 \\ 0 & \overline{\lambda} \end{pmatrix}.$$

Since the eigenvalues of B and C are identical, Theorem 11.1.1 implies that there is a 2×2 complex matrix T such that

$$C = T^{-1}BT. \tag{11.1.4}$$

Moreover, the columns of T are the complex eigenvectors v_1 and v_2 associated to the eigenvalues λ and $\overline{\lambda}$.

It can be checked that the eigenvectors of B are $v_1 = (1,-i)^t$ and $v_2 = (1,i)^t$. On setting

$$T = \begin{pmatrix} 1 & 1 \\ -i & i \end{pmatrix},$$

it is a straightforward calculation to verify that $C = T^{-1}BT$.

As a second example, consider the matrix

$$A = \begin{pmatrix} 4 & 2 & 1 \\ 2 & -3 & 1 \\ 1 & -1 & -3 \end{pmatrix}. \tag{11.1.5*}$$

Using MATLAB we find the eigenvalues of A by typing `eig(A)`. They are:

```
ans =
    4.6432
  -3.3216 + 0.9014i
  -3.3216 - 0.9014i
```

We can diagonalize (over the complex numbers) using MATLAB — indeed MATLAB is programmed to do these calculations over the complex numbers. Type `[T,D] = eig(A)` and obtain

```
T =
   0.9604    -0.1299 + 0.1587i  -0.1299 - 0.1587i
   0.2632     0.0147 - 0.5809i   0.0147 + 0.5809i
   0.0912     0.7788 - 0.1173i   0.7788 + 0.1173i

D =
   4.6432          0                  0
        0    -3.3216 + 0.9014i        0
        0          0            -3.3216 - 0.9014i
```

This calculation can be checked by typing `inv(T)*A*T` to see that the diagonal matrix `D` appears. One can also check that the columns of `T` are eigenvectors of A.

Note that the development here does not depend on the matrix A having real entries. Indeed, this diagonalization can be completed using $n \times n$ matrices with complex entries — and MATLAB can handle such calculations.

Geometric Normal Forms: Block Diagonalization There is a second normal form theorem based on the geometry of rotations and dilatations for real $n \times n$ matrices A. In this normal form we determine all matrices A that have distinct eigenvalues — up to similarity by real $n \times n$ matrices S. The normal form results in matrices that are block diagonal with either 1×1 blocks or 2×2 blocks of the form (11.1.1) on the diagonal.

A real $n \times n$ matrix is in *real block diagonal form* if it is a block diagonal matrix

$$\begin{pmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_m \end{pmatrix}, \tag{11.1.6}$$

where each B_j is either a 1×1 block

$$B_j = \lambda_j$$

for some real number λ_j or a 2×2 block

$$B_j = \begin{pmatrix} \sigma_j & -\tau_j \\ \tau_j & \sigma_j \end{pmatrix} \tag{11.1.7}$$

where σ_j and $\tau_j \neq 0$ are real numbers. A matrix is *real block diagonalizable* if it is similar to a real block diagonal form matrix.

Note that the real eigenvalues of a real block diagonal form matrix are just the real numbers λ_j that occur in the 1×1 blocks. The complex eigenvalues are the eigenvalues of the 2×2 blocks B_j and are $\sigma_j \pm i\tau_j$.

Theorem 11.1.2. *Every $n \times n$ matrix A with n distinct eigenvalues is real block diagonalizable.*

We need two preliminary results.

Lemma 11.1.3. *Let $\lambda_1, \ldots, \lambda_q$ be distinct (possible complex) eigenvalues of an $n \times n$ matrix A. Let v_j be a (possibly complex) eigenvector associated with the eigenvalue λ_j. Then $v_1, \ldots, v_q$ are linearly independent in the sense that if*

$$\alpha_1 v_1 + \cdots + \alpha_q v_q = 0 \tag{11.1.8}$$

for (possibly complex) scalars α_j, then $\alpha_j = 0$ for all j.

Proof The proof is identical in spirit with the proof of Lemma 7.3.2. Proceed by induction on q. When $q = 1$ the lemma is trivially valid, as $\alpha v = 0$ for $v \neq 0$ implies that $\alpha = 0$, even when $\alpha \in \mathbb{C}$ and $v \in \mathbb{C}^n$.

By induction assume the lemma is valid for $q - 1$. Now apply A to (11.1.8) obtaining

$$\alpha_1 \lambda_1 v_1 + \cdots + \alpha_q \lambda_q v_q = 0.$$

Subtract this identity from λ_q times (11.1.8), and obtain

$$\alpha_1(\lambda_1 - \lambda_q)v_1 + \cdots + \alpha_{q-1}(\lambda_{q-1} - \lambda_q)v_{q-1} = 0.$$

By induction

$$\alpha_j(\lambda_j - \lambda_q) = 0$$

for $j = 1, \ldots, q-1$. Since the λ_j are distinct it follows that $\alpha_j = 0$ for $j = 1, \ldots, q-1$. Hence (11.1.8) implies that $\alpha_q v_q = 0$; since $v_q \neq 0$, $\alpha_q = 0$. ■

Lemma 11.1.4. *Let $\mu_1, \ldots, \mu_k$ be distinct real eigenvalues of an $n \times n$ matrix A and let $\nu_1, \overline{\nu}_1 \ldots, \nu_\ell, \overline{\nu}_\ell$ be distinct complex conjugate eigenvalues of A. Let $v_j \in \mathbb{R}^n$ be eigenvectors associated to μ_j and let $w_j = w_j^r + i w_j^i$ be eigenvectors associated with the eigenvalues ν_j. Then the $k + 2\ell$ vectors*

$$v_1, \ldots, v_k, w_1^r, w_1^i, \ldots, w_\ell^r, w_\ell^i$$

in $\mathbb{R}^n$ are linearly independent.

Proof Let $w = w^r + iw^i$ be a vector in $\mathbb{C}^n$ and let β^r and β^i be real scalars. Then

$$\beta^r w^r + \beta^i w^i = \beta w + \overline{\beta}\overline{w}, \tag{11.1.9}$$

where $\beta = \frac{1}{2}(\beta^r - i\beta^i)$. Identity (11.1.9) is verified by direct calculation.

Suppose now that

$$\alpha_1 v_1 + \cdots + \alpha_k v_k + \beta_1^r w_1^r + \beta_1^i w_1^i + \cdots + \beta_\ell^r w_\ell^r + \beta_\ell^i w_\ell^i = 0 \tag{11.1.10}$$

for real scalars α_j, β_j^r and β_j^i. Using (11.1.9) we can rewrite (11.1.10) as

$$\alpha_1 v_1 + \cdots + \alpha_k v_k + \beta_1 w_1 + \overline{\beta}_1 \overline{w}_1 + \cdots + \beta_\ell w_\ell + \overline{\beta}_\ell \overline{w}_\ell = 0,$$

where $\beta_j = \frac{1}{2}(\beta_j^r - i\beta_j^i)$. Since the eigenvalues

$$\mu_1, \ldots, \mu_k, \nu_1, \overline{\nu}_1 \ldots, \nu_\ell, \overline{\nu}_\ell$$

are all distinct, we may apply Lemma 11.1.3 to conclude that $\alpha_j = 0$ and $\beta_j = 0$. It follows that $\beta_j^r = 0$ and $\beta_j^i = 0$, as well, thus proving linear independence. ■

.

Proof of Theorem 11.1.2 Let μ_j for $j = 1, \ldots, k$ be the real eigenvalues of A and let $\nu_j, \overline{\nu}_j$ for $j = 1, \ldots, \ell$ be the complex eigenvalues of A. Since the eigenvalues are all distinct, it follows that $k + 2\ell = n$.

Let v_j and $w_j = w_j^r + iw_j^i$ be eigenvectors associated with the eigenvalues μ_j and $\overline{\nu}_j$. It follows from Lemma 11.1.4 that the n real vectors

$$v_1, \ldots, v_k, w_1^r, w_1^i, \ldots, w_\ell^r, w_\ell^i \tag{11.1.11}$$

are linearly independent and hence form a basis for $\mathbb{R}^n$.

We now show that A is real block diagonalizable. Let S be the $n \times n$ matrix whose columns are the vectors in (11.1.11). Since these vectors are linearly independent, S is invertible. We claim that $S^{-1}AS$ is real block diagonal. This statement is verified by direct calculation.

First, note that $Se_j = v_j$ for $j = 1, \ldots, k$ and compute

$$(S^{-1}AS)e_j = S^{-1}Av_j = \mu_j S^{-1} v_j = \mu_j e_j.$$

It follows that the first k columns of $S^{-1}AS$ are zero except for the diagonal entries, and those diagonal entries equal $\mu_1, \ldots, \mu_k$.

Second, note that $Se_{k+1} = w_1^r$ and $Se_{k+2} = w_1^i$. Write the complex eigenvalues as

$$\nu_j = \sigma_j + i\tau_j.$$

Since $Aw_1 = \overline{\nu}_1 w_1$, it follows that

$$\begin{array}{rcl} Aw_1^r + iAw_1^i & = & (\sigma_1 - i\tau_1)(w_1^r + iw_1^i) \\ & = & (\sigma_1 w_1^r + \tau_1 w_1^i) + i(-\tau_1 w_1^r + \sigma_1 w_1^i). \end{array}$$

Equating real and imaginary parts leads to

$$\begin{array}{rcl} Aw_1^r & = & \sigma_1 w_1^r + \tau_1 w_1^i \\ Aw_1^i & = & -\tau_1 w_1^r + \sigma_1 w_1^i. \end{array} \tag{11.1.12}$$

Using (11.1.12), compute

$$\begin{aligned} (S^{-1}AS)e_{k+1} &= S^{-1}Aw_1^r = S^{-1}(\sigma_1 w_1^r + \tau_1 w_1^i) \\ &= \sigma_1 e_{k+1} + \tau_1 e_{k+2}. \end{aligned}$$

Similarly,

$$\begin{aligned} (S^{-1}AS)e_{k+2} &= S^{-1}Aw_1^i = S^{-1}(-\tau_1 w_1^r + \sigma_1 w_1^i) \\ &= -\tau_1 e_{k+1} + \sigma_1 e_{k+2}. \end{aligned}$$

Thus, the k^{th} and $(k+1)^{st}$ columns of $S^{-1}AS$ have the desired diagonal block in the k^{th} and $(k+1)^{st}$ rows, and have all other entries equal to zero.

The same calculation is valid for the complex eigenvalues $\nu_2, \ldots, \nu_\ell$. Thus, $S^{-1}AS$ is real block diagonal, as claimed. ■

MATLAB Calculations of Real Block Diagonal Form Let C be the 4×4 matrix

$$C = \begin{pmatrix} 1 & 0 & 2 & 3 \\ 2 & 1 & 4 & 6 \\ -1 & -5 & 1 & 3 \\ 1 & 4 & 7 & 10 \end{pmatrix}. \qquad (11.1.13^*)$$

Using MATLAB enter C by typing `e13_2_14` and find the eigenvalues of C by typing `eig(C)` to obtain

```
ans =
   0.5855 + 0.8861i
   0.5855 - 0.8861i
  -0.6399
  12.4690
```

We see that C has two real and two complex conjugate eigenvalues. To find the complex eigenvectors associated with these eigenvalues, type

```
[T,D] = eig(C)
```

MATLAB responds with

```
T =
-0.0787+0.0899i -0.0787-0.0899i  0.0464   0.2209
 0.0772+0.2476i  0.0772-0.2476i  0.0362   0.4803
-0.5558-0.5945i -0.5558+0.5945i -0.8421  -0.0066
 0.3549+0.3607i  0.3549-0.3607i  0.5361   0.8488


D =
0.586+0.886i 0            0          0
0            0.586-0.886i 0        0
0            0            -0.640     0
0.           0               0      12.469
```

The 4×4 matrix T has the eigenvectors of C as columns. The j^{th} column is the eigenvector associated with the j^{th} diagonal entry in the diagonal matrix D.

To find the matrix S that puts C in real block diagonal form, we need to take the real and imaginary parts of the eigenvectors corresponding to the complex eigenvalues and the real eigenvectors corresponding to the real eigenvalues. In this case, type

```
S = [real(T(:,1)) imag(T(:,1)) T(:,3) T(:,4)]
```

to obtain

```
S =
   -0.0787    0.0899    0.0464    0.2209
    0.0772    0.2476    0.0362    0.4803
   -0.5558   -0.5945   -0.8421   -0.0066
    0.3549    0.3607    0.5361    0.8488
```

Note that the 1^{st} and 2^{nd} columns are the real and imaginary parts of the complex eigenvector. Check that `inv(S)*C*S` is the matrix in complex diagonal form

```
ans =
    0.5855    0.8861    0.0000    0.0000
   -0.8861    0.5855    0.0000   -0.0000
    0.0000    0.0000   -0.6399    0.0000
   -0.0000   -0.0000   -0.0000   12.4690
```

as proved in Theorem 11.1.2.

Exercises

11.2 Multiplicity and Generalized Eigenvectors

The difficulty in generalizing the results in the previous two sections to matrices with multiple eigenvalues stems from the fact that these matrices may not have enough (linearly independent) eigenvectors. In this section we present the basic examples of matrices with a deficiency of eigenvectors, as well as the definitions of algebraic and geometric multiplicity. These matrices will be the building blocks of the Jordan normal form theorem — the theorem that classifies all matrices up to similarity.

Deficiency in Eigenvectors for Real Eigenvalues An example of deficiency in eigenvectors is given by the following $n \times n$ matrix

$$J_n(\lambda_0) = \begin{pmatrix} \lambda_0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \lambda_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & \lambda_0 \end{pmatrix} \tag{11.2.1}$$

where $\lambda_0 \in \mathbb{R}$. Note that $J_n(\lambda_0)$ has all diagonal entries equal to λ_0, all superdiagonal entries equal to 1, and all other entries equal to 0. Since $J_n(\lambda_0)$ is upper triangular, all n eigenvalues of $J_n(\lambda_0)$ are equal to λ_0. However, $J_n(\lambda_0)$ has only one linearly independent eigenvector. To verify this assertion let

$$N = J_n(\lambda_0) - \lambda_0 I_n.$$

Then v is an eigenvector of $J_n(\lambda_0)$ if and only if $Nv = 0$. Therefore, $J_n(\lambda_0)$ has a unique linearly independent eigenvector if

Lemma 11.2.1. $\mathrm{nullity}(N) = 1$.

Proof In coordinates the equation $Nv = 0$ is:

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-1} \\ v_n \end{pmatrix} = \begin{pmatrix} v_2 \\ v_3 \\ v_4 \\ \vdots \\ v_n \\ 0 \end{pmatrix} = 0.$$

Thus $v_2 = v_3 = \cdots v_n = 0$, and the solutions are all multiples of e_1. Therefore, the nullity of N is 1. ■

Note that we can express matrix multiplication by N as

$$\begin{aligned} Ne_1 &= 0 \\ Ne_j &= e_{j-1} \qquad j = 2, \ldots, n. \end{aligned} \tag{11.2.2}$$

Note that (11.2.2) implies that $N^n = 0$.

The $n \times n$ matrix N motivates the following definitions.

Definition 11.2.2. Let λ_0 be an eigenvalue of A. The *algebraic multiplicity* of λ_0 is the number of times that λ_0 appears as a root of the characteristic polynomial $p_A(\lambda)$. The *geometric multiplicity* of λ_0 is the number of linearly independent eigenvectors of A having eigenvalue equal to λ_0.

Abstractly, the geometric multiplicity is:

$$\mathrm{nullity}(A - \lambda_0 I_n).$$

Our previous calculations show that the matrix $J_n(\lambda_0)$ has an eigenvalue λ_0 with algebraic multiplicity equal to n and geometric multiplicity equal to 1.

Lemma 11.2.3. *The algebraic multiplicity of an eigenvalue is greater than or equal to its geometric multiplicity.*

Proof For ease of notation we prove this lemma only for real eigenvalues, though the proof for complex eigenvalues is similar. Let A be an $n \times n$ matrix and let λ_0

be a real eigenvalue of A. Let k be the geometric multiplicity of λ_0 and let $v_1, \ldots, v_k$ be k linearly independent eigenvectors of A with eigenvalue λ_0. We can extend $\{v_1, \ldots, v_k\}$ to be a basis $\mathcal{V} = \{v_1, \ldots, v_n\}$ of $\mathbb{R}^n$. In this basis, the matrix of A is

$$[A]_{\mathcal{V}} = \begin{pmatrix} \lambda_0 I_k & (*) \\ 0 & B \end{pmatrix}.$$

The matrices A and $[A]_{\mathcal{V}}$ are similar matrices. Therefore, they have the same characteristic polynomials and the same eigenvalues with the same algebraic multiplicities. It follows from Lemma 7.1.9 that the characteristic polynomial of A is:

$$p_A(\lambda) = p_{[A]_{\mathcal{V}}}(\lambda) = (\lambda - \lambda_0)^k p_B(\lambda).$$

Hence λ_0 appears as a root of $p_A(\lambda)$ at least k times and the algebraic multiplicity of λ_0 is greater than or equal to k. The same proof works when λ_0 is a complex eigenvalue — but all vectors chosen must be complex rather than real. ■

Deficiency in Eigenvectors with Complex Eigenvalues An example of a real matrix with complex conjugate eigenvalues having geometric multiplicity less than algebraic multiplicity is the $2n \times 2n$ block matrix

$$\widehat{J}_n(\lambda_0) = \begin{pmatrix} B & I_2 & 0 & \cdots & 0 & 0 \\ 0 & B & I_2 & \cdots & 0 & 0 \\ 0 & 0 & B & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & B & I_2 \\ 0 & 0 & 0 & \cdots & 0 & B \end{pmatrix} \tag{11.2.3}$$

where $\lambda_0 = \sigma + i\tau$ and B is the 2×2 matrix

$$B = \begin{pmatrix} \sigma & -\tau \\ \tau & \sigma \end{pmatrix}.$$

Lemma 11.2.4. *Let λ_0 be a complex number. Then the algebraic multiplicity of the eigenvalue λ_0 in the $2n \times 2n$ matrix $\widehat{J}_n(\lambda_0)$ is n and the geometric multiplicity is* 1.

Proof We begin by showing that the eigenvalues of $J = \widehat{J}_n(\lambda_0)$ are λ_0 and $\overline{\lambda_0}$, each with algebraic multiplicity n. The characteristic polynomial of J is $p_J(\lambda) = \det(J - \lambda I_{2n})$. From Lemma 7.1.9 of Chapter 7 and induction, we see that $p_J(\lambda) = p_B(\lambda)^n$. Since the eigenvalues of B are λ_0 and $\overline{\lambda_0}$, we have proved that the algebraic multiplicity of each of these eigenvalues in J is n.

Next, we compute the eigenvectors of J. Let $Jv = \lambda_0 v$ and let $v = (v_1, \ldots, v_n)$ where each $v_j \in \mathbb{C}^2$. Observe that $(J - \lambda_0 I_{2n})v = 0$ if and only if

$$\begin{array}{rcl} Qv_1 + v_2 & = & 0 \\ & \vdots & \\ Qv_{n-1} + v_n & = & 0 \\ Qv_n & = & 0, \end{array}$$

where $Q = B - \lambda_0 I_2$. Using the fact that $\lambda_0 = \sigma + i\tau$, it follows that

$$Q = B - \lambda_0 I_2 = -\tau \begin{pmatrix} i & 1 \\ -1 & i \end{pmatrix}.$$

Hence

$$Q^2 = 2\tau^2 i \begin{pmatrix} i & 1 \\ -1 & i \end{pmatrix} = -2\tau i Q.$$

Thus

$$0 = Q^2 v_{n-1} + Qv_n = -2\tau i Q v_{n-1},$$

from which it follows that $Qv_{n-1} + v_n = v_n = 0$. Similarly, $v_2 = \cdots = v_{n-1} = 0$. Since there is only one nonzero complex vector v_1 (up to a complex scalar multiple) satisfying

$$Qv_1 = 0,$$

it follows that the geometric multiplicity of λ_0 in the matrix $\widehat{J}_n(\lambda_0)$ equals 1. ■

Definition 11.2.5. The real matrices $J_n(\lambda_0)$ when $\lambda_0 \in \mathbb{R}$ and $\widehat{J}_n(\lambda_0)$ when $\lambda_0 \in \mathbb{C}$ are *real Jordan blocks*. The matrices $J_n(\lambda_0)$ when $\lambda_0 \in \mathbb{C}$ are (complex) *Jordan blocks*.

Generalized Eigenvectors and Generalized Eigenspaces

What happens when $n \times n$ matrices have fewer that n linearly independent eigenvectors? Answer: The matrices gain generalized eigenvectors.

Definition 11.2.6. A vector $v \in \mathbb{C}^n$ is a *generalized eigenvector* for the $n \times n$ matrix A with eigenvalue λ if

$$(A - \lambda I_n)^k v = 0 \tag{11.2.4}$$

for some positive integer k. The smallest integer k for which (11.2.4) is satisfied is called the *index* of the generalized eigenvector v.

Note: Eigenvectors are generalized eigenvectors with index equal to 1.

Let λ_0 be a real number and let $N = J_n(\lambda_0) - \lambda_0 I_n$. Recall that (11.2.2) implies that $N^n = 0$. Hence every vector in $\mathbb{R}^n$ is a generalized eigenvector for the matrix $J_n(\lambda_0)$. So $J_n(\lambda_0)$ provides a good example of a matrix whose lack of eigenvectors (there is only one independent eigenvector) is made up for by generalized eigenvectors (there are n independent generalized eigenvectors).

Let λ_0 be an eigenvalue of the $n \times n$ matrix A and let $A_0 = A - \lambda_0 I_n$. For simplicity, assume that λ_0 is real. Note that

$$\begin{aligned} \text{null space}(A_0) &\subset \text{null space}(A_0^2) \subset \cdots \\ &\subset \text{null space}(A_0^k) \subset \cdots \\ &\subset \mathbb{R}^n. \end{aligned}$$

Therefore, the dimensions of the null spaces are bounded above by n and there must be a smallest k such that

$$\dim \text{null space}(A_0^k) = \dim \text{null space}(A_0^{k+1}).$$

It follows that

$$\text{null space}(A_0^k) = \text{null space}(A_0^{k+1}). \tag{11.2.5}$$

Lemma 11.2.7. *Let λ_0 be a real eigenvalue of the $n \times n$ matrix A and let $A_0 = A - \lambda_0 I_n$. Let k be the smallest integer for which (11.2.5) is valid. Then*

$$\text{null space}(A_0^k) = \text{null space}(A_0^{k+j})$$

for every interger $j > 0$.

Proof We can prove the lemma by induction on j if we can show that

$$\text{null space}(A_0^{k+1}) = \text{null space}(A_0^{k+2}).$$

Since null space$(A_0^{k+1}) \subset$ null space(A_0^{k+2}), we need to show that

$$\text{null space}(A_0^{k+2}) \subset \text{null space}(A_0^{k+1}).$$

Let $w \in$ null space(A_0^{k+2}). It follows that

$$A^{k+1}Aw = A^{k+2}w = 0;$$

so $Aw \in$ null space(A_0^{k+1}) = null space(A_0^k), by (11.2.5). Therefore,

$$A^{k+1}w = A^k(Aw) = 0,$$

which verifies that $w \in$ null space(A_0^{k+1}). ■

Let V_{λ_0} be the set of all generalized eigenvectors of A with eigenvalue λ_0. Let k be the smallest integer satisfying (11.2.5), then Lemma 11.2.7 implies that

$$V_{\lambda_0} = \text{null space}(A_0^k) \subset \mathbb{R}^n$$

is a subspace called the *generalized eigenspace* of A associated to the eigenvalue λ_0. It will follow from the Jordan normal form theorem (see Theorem 11.3.2) that the dimension of V_{λ_0} is the algebraic multiplicity of λ_0.

An Example of Generalized Eigenvectors Find the generalized eigenvectors of the 4×4 matrix

$$A = \begin{pmatrix} -24 & -58 & -2 & -8 \\ 15 & 35 & 1 & 4 \\ 3 & 5 & 7 & 4 \\ 3 & 6 & 0 & 6 \end{pmatrix}. \tag{11.2.6*}$$

and their indices. When finding generalized eigenvectors of a matrix A, the first two steps are:

(i) Find the eigenvalues of A.

(ii) Find the eigenvectors of A.

After entering A into MATLAB by typing `e13_3_6`, we type `eig(A)` and find that all of the eigenvalues of A equal 6. Without additional information, there could be 1,2,3 or 4 linearly independent eigenvectors of A corresponding to the eigenvalue 6. In MATLAB we determine the number of linearly independent eigenvectors by typing `null(A-6*eye(4))` and obtaining

```
ans =
    0.8892         0
   -0.4446    0.0000
   -0.0262    0.9701
   -0.1046   -0.2425
```

We now know that (numerically) there are two linearly independent eigenvectors. The next step is find the number of independent generalized eigenvectors of index 2. To complete this calculation, we find a basis for the null space of $(A - 6I_4)^2$ by typing `null((A-6*eye(4))^2)` obtaining

```
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

Thus, for this example, all generalized eigenvectors that are not eigenvectors have index 2.

Exercises

11.3 The Jordan Normal Form Theorem

The question that we discussed in Sections 7.3 and 11.1 is: Up to similarity, what is the simplest form that a matrix can have? We have seen that if A has real distinct eigenvalues, then A is real diagonalizable. That is, A is similar to a diagonal matrix whose diagonal entries are the real eigenvalues of A. Similarly, if A has distinct real and complex eigenvalues, then A is complex diagonalizable; that is, A is similar either to a diagonal matrix whose diagonal entries are the real and complex eigenvalues of A or to a real block diagonal matrix.

In this section we address the question of simplest form when a matrix has multiple eigenvalues. In much of this discussion we assume that A is an $n \times n$ matrix with only real eigenvalues. Lemma 7.3.3 shows that if the eigenvectors of A form a basis, then A is diagonalizable. Indeed, for A to be diagonalizable, there must be a basis of eigenvectors of A. It follows that if A is not diagonalizable, then A must have fewer than n linearly independent eigenvectors.

The prototypical examples of matrices having fewer eigenvectors than eigenvalues are the matrices $J_n(\lambda)$ for λ real (see (11.2.1)) and $\widehat{J}_n(\lambda)$ for λ complex (see (11.2.3)).

Definition 11.3.1. A matrix is in *Jordan normal form* if it is block diagonal and the matrix in each block on the diagonal is a Jordan block, that is, $J_\ell(\lambda)$ for some integer ℓ and some real or complex number λ.

A matrix is in *real Jordan normal form* if it is block diagonal and the matrix in each block on the diagonal is a real Jordan block, that is, either $J_\ell(\lambda)$ for some integer ℓ and some real number λ or $\widehat{J}_\ell(\lambda)$ for some integer ℓ and some complex number λ.

The main theorem about Jordan normal form is:

Theorem 11.3.2 (Jordan normal form)**.** *Let A be an $n \times n$ matrix. Then A is similar to a Jordan normal form matrix and to a real Jordan normal form matrix.*

This theorem is proved by constructing a basis $\mathcal{V}$ for $\mathbb{R}^n$ so that the matrix $S^{-1}AS$ is in Jordan normal form, where S is the matrix whose columns consists of vectors in $\mathcal{V}$. The algorithm for finding the basis $\mathcal{V}$ is complicated and is found in Appendix 11.5. In this section we construct $\mathcal{V}$ only in the special and simpler case where each eigenvalue of A is real and is associated with exactly one Jordan block.

More precisely, let $\lambda_1, \ldots, \lambda_s$ be the distinct eigenvalues of A and let
$$A_j = A - \lambda_j I_n.$$
The eigenvectors corresponding to λ_j are the vectors in the null space of A_j and the generalized eigenvectors are the vectors in the null space of A_j^k for some k. The dimension of the null space of A_j is precisely the number of Jordan blocks of A associated to the eigenvalue λ_j. So the assumption that we make here is
$$\mathrm{nullity}(A_j) = 1$$
for $j = 1, \ldots, s$.

Let k_j be the integer whose existence is specified by Lemma 11.2.7. Since, by assumption, there is only one Jordan block associated with the eigenvalue λ_j, it follows that k_j is the algebraic multiplicity of the eigenvalue λ_j.

To find a basis in which the matrix A is in Jordan normal form, we proceed as follows. First, let w_{jk_j} be a vector in
$$\text{null space}(A_j^{k_j}) - \text{null space}(A_j^{k_j-1}).$$
Define the vectors w_{ji} by
$$\begin{array}{rcl} w_{j,k_j-1} & = & A_j w_{j,k_j} \\ & \vdots & \\ w_{j,1} & = & A_j w_{j,2}. \end{array}$$

Second, when λ_j is real, let the k_j vectors $v_{ji} = w_{ji}$, and when λ_j is complex, let the $2k_j$ vectors v_{ji} be defined by

$$\begin{array}{rcl} v_{j,2i-1} & = & \mathrm{Re}(w_{ji}) \\ v_{j,2i} & = & \mathrm{Im}(w_{ji}). \end{array}$$

Let $\mathcal{V}$ be the set of vectors $v_{ji} \in \mathbb{R}^n$. We will show in Appendix 11.5 that the set $\mathcal{V}$ consists of n vectors and is a basis of $\mathbb{R}^n$. Let S be the matrix whose columns are the vectors in $\mathcal{V}$. Then $S^{-1}AS$ is in Jordan normal form.

The Cayley Hamilton Theorem As a corollary of the Jordan normal form theorem, we prove the Cayley Hamilton theorem which states that a *square matrix satisfies its characteristic polynomial*. More precisely:

Theorem 11.3.3 (Cayley Hamilton)**.** *Let A be a square matrix and let $p_A(\lambda)$ be its characteristic polynomial. Then*

$$p_A(A) = 0.$$

Proof Let A be an $n \times n$ matrix. The characteristic polynomial of A is

$$p_A(\lambda) = \det(A - \lambda I_n).$$

Suppose that $B = P^{-1}AP$ is a matrix similar to A. Theorem 7.2.8 states that $p_B = p_A$. Therefore

$$p_B(B) = p_A(P^{-1}AP) = P^{-1}p_A(A)P.$$

So if the Cayley Hamilton theorem holds for a matrix similar to A, then it is valid for the matrix A. Moreover, using the Jordan normal form theorem, we may assume that A is in Jordan normal form.

Suppose that A is block diagonal, that is

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix},$$

where A_1 and A_2 are square matrices. Then

$$p_A(\lambda) = p_{A_1}(\lambda)p_{A_2}(\lambda).$$

This observation follows directly from Lemma 7.1.9. Since

$$A^k = \begin{pmatrix} A_1^k & 0 \\ 0 & A_2^k \end{pmatrix},$$

it follows that

$$\begin{aligned} p_A(A) &= \begin{pmatrix} p_A(A_1) & 0 \\ 0 & p_A(A_2) \end{pmatrix} \\ &= \begin{pmatrix} p_{A_1}(A_1)p_{A_2}(A_1) & 0 \\ 0 & p_{A_1}(A_2)p_{A_2}(A_2) \end{pmatrix}. \end{aligned}$$

It now follows from this calculation that if the Cayley Hamilton theorem is valid for Jordan blocks, then $p_{A_1}(A_1) = 0 = p_{A_2}(A_2)$. So $p_A(A) = 0$ and the Cayley Hamilton theorem is valid for all matrices.

A direct calculation shows that Jordan blocks satisfy the Cayley Hamilton theorem. To begin, suppose that the eigenvalue of the Jordan block is real. Note that the characteristic polynomial of the Jordan block $J_n(\lambda_0)$ in (11.2.1) is $(\lambda - \lambda_0)^n$. Indeed, $J_n(\lambda_0) - \lambda_0 I_n$ is strictly upper triangular and $(J_n(\lambda_0) - \lambda_0 I_n)^n = 0$. If λ_0 is complex, then either repeat this calculation using the complex Jordan form or show by direct calculation that $(A - \lambda_0 I_n)(A - \overline{\lambda_0} I_n)$ is strictly upper triangular when $A = \widehat{J}_n(\lambda_0)$ is the real Jordan form of the Jordan block in (11.2.3). ■

An Example Consider the 4×4 matrix

$$A = \begin{pmatrix} -147 & -106 & -66 & -488 \\ 604 & 432 & 271 & 1992 \\ 621 & 448 & 279 & 2063 \\ -169 & -122 & -76 & -562 \end{pmatrix}. \qquad (11.3.1^*)$$

Using MATLAB we can compute the characteristic polynomial of A by typing

```
poly(A)
```

The output is

```
ans =
  1.0000   -2.0000  -15.0000   -0.0000   -0.0000
```

Note that since A is a matrix of integers we know that the coefficients of the characteristic polynomial of A must be integers. Thus the characteristic polynomial is exactly:

$$p_A(\lambda) = \lambda^4 - 2\lambda^3 - 15\lambda^2 = \lambda^2(\lambda - 5)(\lambda + 3).$$

So $\lambda_1 = 0$ is an eigenvalue of A with algebraic multiplicity two and $\lambda_2 = 5$ and $\lambda_3 = -3$ are simple eigenvalues of multiplicity one.

We can find eigenvectors of A corresponding to the simple eigenvalues by typing

```
v2 = null(A-5*eye(4));
v3 = null(A+3*eye(4));
```

At this stage we do not know how many linearly independent eigenvectors have eigenvalue 0. There are either one or two linearly independent eigenvectors and we determine which by typing `null(A)` and obtaining

```
ans =
  -0.1818
   0.6365
   0.7273
  -0.1818
```

So MATLAB tells us that there is just one linearly independent eigenvector having 0 as an eigenvalue. There must be a generalized eigenvector in V_0. Indeed, the null space of A^2 is two dimensional and this fact can be checked by typing

```
null2 = null(A^2)
```

obtaining

```
null2 =
    0.2193   -0.2236
   -0.5149   -0.8216
   -0.8139    0.4935
    0.1561    0.1774
```

Choose one of these vectors, say the first vector, to be v_{12} by typing

```
v12 = null2(:,1);
```

Since the algebraic multiplicity of the eigenvalue 0 is two, we choose the fourth basis vector be $v_{11} = Av_{12}$. In MATLAB we type

```
v11 = A*v12
```

obtaining

```
v11 =
   -0.1263
    0.4420
    0.5051
   -0.1263
```

Since `v11` is nonzero, we have found a basis for V_0. We can now put the matrix A in Jordan normal form by setting

```
S = [v11 v12 v2 v3];
J = inv(S)*A*S
```

to obtain

```
J =
   -0.0000    1.0000    0.0000   -0.0000
    0.0000    0.0000    0.0000   -0.0000
   -0.0000   -0.0000    5.0000    0.0000
    0.0000   -0.0000   -0.0000   -3.0000
```

We have only discussed a Jordan normal form example when the eigenvalues are real and multiple. The case when the eigenvalues are complex and multiple first occurs when $n = 4$. A sample complex Jordan block when the matrix has algebraic multiplicity two eigenvalues $\sigma \pm i\tau$ of geometric multiplicity one is

$$\begin{pmatrix} \sigma & -\tau & 1 & 0 \\ \tau & \sigma & 0 & 1 \\ 0 & 0 & \sigma & -\tau \\ 0 & 0 & \tau & \sigma \end{pmatrix}.$$

Numerical Difficulties When a matrix has multiple eigenvalues, then numerical difficulties can arise when using the MATLAB command `eig(A)`, as we now explain.

Let $p(\lambda) = \lambda^2$. Solving $p(\lambda) = 0$ is very easy — in theory — as $\lambda = 0$ is a double root of p. Suppose, however, that we want to solve $p(\lambda) = 0$ numerically. Then, numerical errors will lead to solving the equation

$$\lambda^2 = \epsilon$$

where ϵ is a small number. Note that if $\epsilon > 0$, the solutions are $\pm\sqrt{\epsilon}$; while if $\epsilon < 0$, the solutions are $\pm i\sqrt{|\epsilon|}$. Since numerical errors are machine dependent, ϵ can be of either sign. The numerical process of finding double roots of a characteristic polynomial (that is, double eigenvalues of a matrix) is similar to numerically solving the equation $\lambda^2 = 0$, as we shall see.

For example, on a *Sun SPARCstation 10* using MATLAB version 4.2c, the eigenvalues of the 4×4 matrix A in (11.3.1*) (in `format long`) obtained using `eig(A)` are:

```
ans =
  5.00000000001021
 -0.00000000000007 + 0.00000023858927i
 -0.00000000000007 - 0.00000023858927i
 -3.00000000000993
```

That is, MATLAB computes two complex conjugate eigenvalues

$$\pm 0.00000023858927i$$

which corresponds to an ϵ of `-5.692483975913288e-14`. On a *IBM* compatible 486 computer using MATLAB version 4.2 the same computation yields eigenvalues

```
ans=
 4.99999999999164
 0.00000057761008
-0.00000057760735
-2.99999999999434
```

That is, on this computer MATLAB computes two real, near zero, eigenvalues

$$\pm 0.00000057761$$

that corresponds to an ϵ of `3.336333121e-13`. These errors are within round off error in double precision computation.

A consequence of these kinds of error, however, is that when a matrix has multiple eigenvalues, we cannot use the command `[V,D] = eig(A)` with confidence. On the *Sun SPARCstation*, this command yields a matrix

```
V =
  -0.1652            0.0000 - 0.1818i   0.0000 + 0.1818i  -0.1642
   0.6726           -0.0001 + 0.6364i  -0.0001 - 0.6364i   0.6704
   0.6962           -0.0001 + 0.7273i  -0.0001 - 0.7273i   0.6978
  -0.1888            0.0000 - 0.1818i   0.0000 + 0.1818i  -0.1915
```

that suggests that A has two complex eigenvectors corresponding to the 'complex' pair of near zero eigenvalues. The *IBM* compatible yields the matrix

```
V =
   -0.1652    0.1818   -0.1818   -0.1642
    0.6726   -0.6364    0.6364    0.6704
    0.6962   -0.7273    0.7273    0.6978
   -0.1888    0.1818   -0.1818   -0.1915
```

indicating that MATLAB has found two real eigenvectors corresponding to the near zero real eigenvalues. Note that the two eigenvectors corresponding to the eigenvalues 5 and -3 are correct on both computers.

Exercises

11.4 *Markov Matrix Theory

In this appendix we use the Jordan normal form theorem to study the asymptotic dynamics of transition matrices such as those of Markov chains introduced in Section 4.8.

The basic result is the following theorem.

Theorem 11.4.1. *Let A be an $n \times n$ matrix and assume that all eigenvalues λ of A satisfy $|\lambda| < 1$. Then for every vector $v_0 \in \mathbb{R}^n$*

$$\lim_{k\to\infty} A^k v_0 = 0. \tag{11.4.1}$$

Proof Suppose that A and B are similar matrices; that is, $B = SAS^{-1}$ for some invertible matrix S. Then $B^k = SA^kS^{-1}$ and for any vector $v_0 \in \mathbb{R}^n$ (11.4.1) is valid if and only if

$$\lim_{k\to\infty} B^k v_0 = 0.$$

Thus, when proving this theorem, we may assume that A is in Jordan normal form.

Suppose that A is in block diagonal form; that is, suppose

$$A = \begin{pmatrix} C & 0 \\ 0 & D \end{pmatrix},$$

where C is an $\ell \times \ell$ matrix and D is a $(n-\ell)\times(n-\ell)$ matrix. Then

$$A^k = \begin{pmatrix} C^k & 0 \\ 0 & D^k \end{pmatrix}.$$

So for every vector $v_0 = (w_0, u_0) \in \mathbb{R}^\ell \times \mathbb{R}^{n-\ell}$ (11.4.1) is valid if and only if

$$\lim_{k\to\infty} C^k v_0 = 0 \quad \text{and} \quad \lim_{k\to\infty} D^k v_0 = 0.$$

So, when proving this theorem, we may assume that A is a Jordan block.

Consider the case of a simple Jordan block. Suppose that $n = 1$ and that $A = (\lambda)$ where λ is either real or complex. Then

$$A^k v_0 = \lambda^k v_0.$$

It follows that (11.4.1) is valid precisely when $|\lambda| < 1$. Next, suppose that A is a nontrivial Jordan block. For example, let

$$A = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} = \lambda I_2 + N$$

where $N^2 = 0$. It follows by induction that

$$A^k v_0 = \lambda^k v_0 + k\lambda^{k-1} N v_0 = \lambda^k v_0 + k\lambda^k \frac{1}{\lambda} N v_0.$$

Thus (11.4.1) is valid precisely when $|\lambda| < 1$. The reason for this convergence is as follows. The first term converges to 0 as before but the second term is the product of three terms k, λ^k, and $\frac{1}{\lambda} N v_0$. The first increases to infinity, the second decreases to zero, and the third is constant independent of k. In fact, geometric decay (λ^k, when $|\lambda| < 1$) always beats polynomial growth. Indeed,

$$\lim_{m\to\infty} m^j \lambda^m = 0 \tag{11.4.2}$$

for any integer j. This fact can be proved using l'Hôspital's rule and induction.

So we see that when A has a nontrivial Jordan block, convergence is subtler than when A has only simple Jordan blocks, as initially the vectors Av_0 grow in magnitude. For example, suppose that $\lambda = 0.75$ and $v_0 = (1,0)^t$. Then $A^8 v_0 = (0.901, 0.075)^t$ is the first vector in the sequence $A^k v_0$ whose norm is less than 1; that is, $A^8 v_0$ is the first vector in the sequence closer to the origin than v_0.

It is also true that (11.4.1) is valid for any Jordan block A and for all v_0 precisely when $|\lambda| < 1$. To verify this fact we use the binomial theorem. We can write a nontrivial

Jordan block as $\lambda I_n + N$ where $N^{k+1} = 0$ for some integer k. We just discussed the case $k = 1$. In this case

$$(\lambda I_n + N)^m = \lambda^m I_n + m\lambda^{m-1}N + \binom{m}{2}\lambda^{m-2}N^2 + \cdots + \binom{m}{k}\lambda^{m-k}N^k,$$

where

$$\binom{m}{j} = \frac{m!}{j!(m-j)!} = \frac{m(m-1)\cdots(m-j+1)}{j!}.$$

To verify that

$$\lim_{m\to\infty}(\lambda I_n + N)^m = 0$$

we need only verify that each term

$$\lim_{m\to\infty}\binom{m}{j}\lambda^{m-j}N^j = 0$$

Such terms are the product of three terms

$$m(m-1)\cdots(m-j+1) \quad \text{and} \quad \lambda^m \quad \text{and} \quad \frac{1}{j!\lambda^j}N^j.$$

The first term has polynomial growth to infinity dominated by m^j, the second term decreases to zero geometrically, and the third term is constant independent of m. The desired convergence to zero follows from (11.4.2). ■

Definition 11.4.2. The $n \times n$ matrix A has a *dominant* eigenvalue $\lambda_0 > 0$ if λ_0 is a simple eigenvalue and all other eigenvalues λ of A satisfy $|\lambda| < \lambda_0$.

Theorem 11.4.3. *Let P be a Markov matrix. Then 1 is a dominant eigenvalue of P.*

Proof Recall from Chapter 3, Definition 4.8.1 that a Markov matrix is a square matrix P whose entries are nonnegative, whose rows sum to 1, and for which a power P^k that has all positive entries. To prove this theorem we must show that all eigenvalues λ of P satisfy $|\lambda| \leq 1$ and that 1 is a simple eigenvalue of P.

Let λ be an eigenvalue of P and let $v = (v_1, \ldots, v_n)^t$ be an eigenvector corresponding to the eigenvalue λ. We prove that $|\lambda| \leq 1$. Choose j so that $|v_j| \geq |v_i|$ for all i. Since $Pv = \lambda v$, we can equate the j^{th} coordinates of both sides of this equality, obtaining

$$p_{j1}v_1 + \cdots + p_{jn}v_n = \lambda v_j.$$

Therefore,

$$|\lambda||v_j| = |p_{j1}v_1 + \cdots + p_{jn}v_n| \leq p_{j1}|v_1| + \cdots + p_{jn}|v_n|,$$

since the p_{ij} are nonnegative. It follows that

$$|\lambda||v_j| \leq (p_{j1} + \cdots + p_{jn})|v_j| = |v_j|,$$

since $|v_i| \leq |v_j|$ and rows of P sum to 1. Since $|v_j| > 0$, it follows that $\lambda \leq 1$.

Next we show that 1 is a simple eigenvalue of P. Recall, or just calculate directly, that the vector $(1, \ldots, 1)^t$ is an eigenvector of P with eigenvalue 1. Now let $v = (v_1, \ldots, v_n)^t$ be an eigenvector of P with eigenvalue 1. Let $Q = P^k$ so that all entries of Q are positive. Observe that v is an eigenvector of Q with eigenvalue 1, and hence that all rows of Q also sum to 1.

To show that 1 is a simple eigenvalue of Q, and therefore of P, we must show that all coordinates of v are equal. Using the previous estimates (with $\lambda = 1$), we obtain

$$|v_j| = |q_{j1}v_1 + \cdots + q_{jn}v_n| \leq q_{j1}|v_1| + \cdots + q_{jn}|v_n| \leq |v_j|. \tag{11.4.3}$$

Hence

$$|q_{j1}v_1 + \cdots + q_{jn}v_n| = q_{j1}|v_1| + \cdots + q_{jn}|v_n|.$$

This equality is valid only if all of the v_i are nonnegative or all are nonpositive. Without loss of generality, we assume that all $v_i \geq 0$. It follows from (11.4.3) that

$$v_j = q_{j1}v_1 + \cdots + q_{jn}v_n.$$

Since $q_{ji} > 0$, this inequality can hold only if all of the v_i are equal. ■

Theorem 11.4.4. *(a) Let Q be an $n \times n$ matrix with dominant eigenvalue $\lambda > 0$ and associated eigenvector v. Let v_0 be any vector in $\mathbb{R}^n$. Then*

$$\lim_{k\to\infty} \frac{1}{\lambda^k} Q^k v_0 = cv,$$

for some scalar c.

(b) Let P be a Markov matrix and v_0 a nonzero vector in $\mathbb{R}^n$ with all entries nonnegative. Then

$$\lim_{k\to\infty} (P^t)^k v_0 = V$$

where V is the eigenvector of P^t with eigenvalue 1 such that the sum of the entries in V is equal to the sum of the entries in v_0.

Proof (a) After a similarity transformation, if needed, we can assume that Q is in Jordan normal form. More precisely, we can assume that

$$\frac{1}{\lambda} Q = \begin{pmatrix} 1 & 0 \\ 0 & A \end{pmatrix}$$

where A is an $(n-1) \times (n-1)$ matrix with all eigenvalues μ satisfying $|\mu| < 1$. Suppose $v_0 = (c_0, w_0) \in \mathbb{R} \times \mathbb{R}^{n-1}$. It follows from Theorem 11.4.1 that

$$\lim_{k\to\infty} \frac{1}{\lambda^k} Q^k v_0 = \lim_{k\to\infty} (\frac{1}{\lambda} Q)^k v_0 = \lim_{k\to\infty} \begin{pmatrix} c_0 & 0 \\ 0 & A^k w_0 \end{pmatrix} = c_0 e_1.$$

Since e_1 is the eigenvector of Q with eigenvalue λ Part (a) is proved.

(b) Theorem 11.4.3 states that a Markov matrix has a dominant eigenvalue equal to 1. The Jordan normal form theorem implies that the eigenvalues of P^t are equal to the eigenvalues of P with the same algebraic and geometric multiplicities. It follows that 1 is also a dominant eigenvalue of P^t. It follows from Part (a) that

$$\lim_{k\to\infty} (P^t)^k v_0 = cV$$

for some scalar c. But Theorem 4.8.3 in Chapter 3 implies that the sum of the entries in v_0 equals the sum of the entries in cV which, by assumption equals the sum of the entries in V. Thus, $c = 1$. ■

Exercises

11.5 *Proof of Jordan Normal Form

We prove the Jordan normal form theorem under the assumption that the eigenvalues of A are all real. The proof for matrices having both real and complex eigenvalues proceeds along similar lines.

Let A be an $n \times n$ matrix, let $\lambda_1, \ldots, \lambda_s$ be the distinct eigenvalues of A, and let $A_j = A - \lambda_j I_n$.

Lemma 11.5.1. *The linear mappings A_i and A_j commute.*

Proof Just compute

$$A_i A_j = (A-\lambda_i I_n)(A-\lambda_j I_n) = A^2 - \lambda_i A - \lambda_j A + \lambda_i \lambda_j I_n,$$

and

$$A_j A_i = (A-\lambda_j I_n)(A-\lambda_i I_n) = A^2 - \lambda_j A - \lambda_i A + \lambda_j \lambda_i I_n.$$

So $A_i A_j = A_j A_i$, as claimed. ■

Let V_j be the generalized eigenspace corresponding to eigenvalue λ_j.

Lemma 11.5.2. *$A_i : V_j \to V_j$ is invertible when $i \neq j$.*

Proof Recall from Lemma 11.2.7 that $V_j = \text{null space}(A_j^k)$ for some $k \geq 1$. Suppose that $v \in V_j$. We first verify that $A_i v$ is also in V_j. Using Lemma 11.5.1, just compute

$$A_j^k A_i v = A_i A_j^k v = A_i 0 = 0.$$

Therefore, $A_i v \in \text{null space}(A_j^k) = V_j$.

Let B be the linear mapping $A_i|V_j$. It follows from Chapter 8, Theorem 8.2.3 that

$$\text{nullity}(B) + \dim \text{range}(B) = \dim(V_j).$$

Now $w \in \text{null space}(B)$ if $w \in V_j$ and $A_i w = 0$. Since $A_i w = (A-\lambda_i I_n)w = 0$, it follows that $Aw = \lambda_i w$. Hence

$$A_j w = (A - \lambda_j I_n)w = (\lambda_i - \lambda_j)w$$

and

$$A_j^k w = (\lambda_i - \lambda_j)^k w.$$

Since $\lambda_i \neq \lambda_j$, it follows that $A_j^k w = 0$ only when $w = 0$. Hence the nullity of B is zero. We conclude that

$$\dim \text{range}(B) = \dim(V_j).$$

Thus, B is invertible, since the domain and range of B are the same space. ■

Lemma 11.5.3. *Nonzero vectors taken from different generalized eigenspaces V_j are linearly independent. More precisely, if $w_j \in V_j$ and*

$$w = w_1 + \cdots + w_s = 0,$$

then $w_j = 0$.

Proof Let $V_j = \text{null space}(A_j^{k_j})$ for some integer k_j. Let $C = A_2^{k_2} \circ \cdots \circ A_s^{k_s}$. Then

$$0 = Cw = Cw_1,$$

since $A_j^{k_j} w_j = 0$ for $j = 2, \ldots, s$. But Lemma 11.5.2 implies that $C|V_1$ is invertible. Therefore, $w_1 = 0$. Similarly, all of the remaining w_j have to vanish. ■

Lemma 11.5.4. *Every vector in $\mathbb{R}^n$ is a linear combination of vectors in the generalized eigenspaces V_j.*

Proof Let W be the subspace of $\mathbb{R}^n$ consisting of all vectors of the form $z_1 + \cdots + z_s$ where $z_j \in V_j$. We need to verify that $W = \mathbb{R}^n$. Suppose that W is a proper subspace. Then choose a basis $w_1, \ldots, w_t$ of W and extend this set to a basis $\mathcal{W}$ of $\mathbb{R}^n$. In this basis the matrix $[A]_{\mathcal{W}}$ has block form, that is,

$$[A]_{\mathcal{W}} = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix},$$

where A_{22} is an $(n-t)\times(n-t)$ matrix. The eigenvalues of A_{22} are eigenvalues of A. Since all of the distinct eigenvalues and eigenvectors of A are accounted for in W (that is, in A_{11}), we have a contradiction. So $W = \mathbb{R}^n$, as claimed. ■

Lemma 11.5.5. *Let $\mathcal{V}_j$ be a basis for the generalized eigenspaces V_j and let $\mathcal{V}$ be the union of the sets $\mathcal{V}_j$. Then $\mathcal{V}$ is a basis for $\mathbb{R}^n$.*

Proof We first show that the vectors in $\mathcal{V}$ span $\mathbb{R}^n$. It follows from Lemma 11.5.4 that every vector in $\mathbb{R}^n$ is a linear combination of vectors in V_j. But each vector in V_j is a linear combination of vectors in $\mathcal{V}_j$. Hence, the vectors in $\mathcal{V}$ span $\mathbb{R}^n$.

Second, we show that the vectors in $\mathcal{V}$ are linearly independent. Suppose that a linear combination of vectors in $\mathcal{V}$ sums to zero. We can write this sum as

$$w_1 + \cdots + w_s = 0,$$

where w_j is the linear combination of vectors in $\mathcal{V}_j$. Lemma 11.5.3 implies that each $w_j = 0$. Since $\mathcal{V}_j$ is a basis for V_j, it follows that the coefficients in the linear combinations w_j must all be zero. Hence, the vectors in $\mathcal{V}$ are linearly independent.

Finally, it follows from Theorem 5.5.3 of Chapter 5 that $\mathcal{V}$ is a basis. ■

Lemma 11.5.6. *In the basis $\mathcal{V}$ of $\mathbb{R}^n$ guaranteed by Lemma 11.5.5, the matrix $[A]_{\mathcal{V}}$ is block diagonal, that is,*

$$[A]_{\mathcal{V}} = \begin{pmatrix} A_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & A_{ss} \end{pmatrix},$$

where all of the eigenvalues of A_{jj} equal λ_j.

Proof It follows from Lemma 11.5.1 that $A : V_j \to V_j$. Suppose that $v_j \in \mathcal{V}_j$. Then Av_j is in V_j and Av_j is a linear combination of vectors in $\mathcal{V}_j$. The block diagonalization of $[A]_{\mathcal{V}}$ follows. Since $V_j = \text{null space}(A_j^{k_j})$, it follows that all eigenvalues of A_{jj} equal λ_j. ■

Lemma 11.5.6 implies that to prove the Jordan normal form theorem, we must find a basis in which the matrix A_{jj} is in Jordan normal form. So, without loss of generality, we may assume that all eigenvalues of A equal λ_0, and then find a basis in which A is in Jordan normal form. Moreover, we can replace A by the matrix $A - \lambda_0 I_n$, a matrix all of whose eigenvalues are zero. So, without loss of generality, we assume that A is an $n\times n$ matrix all of whose eigenvalues are zero. We now sketch the remainder of the proof of Theorem 11.3.2.

Let k be the smallest integer such that $\mathbb{R}^n = \text{null space}(A^k)$ and let

$$s = \dim \text{null space}(A^k) - \dim \text{null space}(A^{k-1}) > 0.$$

Let $z_1, \ldots, z_{n-s}$ be a basis for null space(A^{k-1}) and extend this set to a basis for null space(A^k) by adjoining the linearly independent vectors $w_1, \ldots, w_s$. Let

$$W_k = \text{span}\{w_1, \ldots, w_s\}.$$

It follows that $W_k \cap \text{null space}(A^{k-1}) = \{0\}$.

We claim that the ks vectors $\mathcal{W} = \{w_{j\ell} = A^\ell(w_j)\}$ where $0 \le \ell \le k-1$ and $1 \le j \le s$ are linearly independent. We can write any linear combination of the vectors in $\mathcal{W}$ as $y_k + \cdots + y_1$, where $y_j \in A^{k-j}(W_k)$. Suppose that

$$y_k + \cdots + y_1 = 0.$$

Then $A^{k-1}(y_k + \cdots + y_1) = A^{k-1}y_k = 0$. Therefore, y_k is in W_k and in null space(A^{k-1}). Hence, $y_k = 0$.

Similarly, $A^{k-2}(y_{k-1}+\cdots+y_1) = A^{k-2}y_{k-1} = 0$. But $y_{k-1} = A\hat{y}_k$ where $\hat{y}_k \in W_k$ and $\hat{y}_k \in \text{null space}(A^{k-1})$. Hence, $\hat{y}_k = 0$ and $y_{k-1} = 0$. Similarly, all of the $y_j = 0$. It follows from $y_j = 0$ that a linear combination of the vectors $A^{k-j}(w_1),\ldots,A^{k-j}(w_s)$ is zero; that is

$$0 = \beta_1 A^{k-j}(w_1)+\cdots+\beta_s A^{k-j}(w_s) = A^{k-j}(\beta_1 w_1+\cdots+\beta_s w_s).$$

Applying A^{j-1} to this expression, we see that

$$\beta_1 w_1 + \cdots + \beta_s w_s$$

is in W_k and in the null space(A^{k-1}). Hence,

$$\beta_1 w_1 + \cdots + \beta_s w_s = 0.$$

Since the w_j are linearly independent, each $\beta_j = 0$, thus verifying the claim.

Next, we find the largest integer m so that

$$t = \dim \text{null space}(A^m) - \dim \text{null space}(A^{m-1}) > 0.$$

Proceed as above. Choose a basis for null space(A^{m-1}) and extend to a basis for null space(A^m) by adjoining the vectors $x_1,\ldots,x_t$. Adjoin the mt vectors $A^\ell x_j$ to the set $\mathcal{V}$ and verify that these vectors are all linearly independent. And repeat the process. Eventually, we arrive at a basis for $\mathbb{R}^n = \text{null space}(A^k)$.

In this basis the matrix $[A]_\mathcal{V}$ is block diagonal; indeed, each of the blocks is a Jordan block, since

$$A(w_{j\ell}) = \begin{cases} w_{j(\ell-1)} & 0 < \ell \leq k-1 \\ 0 & \ell = 1 \end{cases}.$$

Note the resemblance with (11.2.2).

12 Matlab Commands

† indicates an `laode` toolbox command not found in MATLAB .

Chapter 1: Preliminaries

Editing and Number Commands

`quit`	Ends MATLAB session
`;`	(a) At end of line the semicolon suppresses echo printing
	(b) When entering an array the semicolon indicates a new row
↑	Displays previous MATLAB command
`[]`	Brackets indicating the beginning and the end of a vector or a matrix
`x=y`	Assigns `x` the value of `y`
`x(j)`	Recalls j^{th} entry of vector x
`A(i,j)`	Recalls i^{th} row, j^{th} column of matrix A
`A(i,:)`	Recalls i^{th} row of matrix A
`A(:,j)`	Recalls j^{th} column of matrix A

Vector Commands

`norm(x)`	The norm or length of a vector x
`dot(x,y)`	Computes the dot product of vectors x and y
†`addvec(x,y)`	Graphics display of vector addition in the plane
†`addvec3(x,y)`	Graphics display of vector addition in three dimensions

Matrix Commands

`A'`	(Conjugate) transpose of matrix
`zeros(m,n)`	Creates an $m \times n$ matrix all of whose entries equal 0
`zeros(n)`	Creates an $n \times n$ matrix all of whose entries equal 0
`diag(x)`	Creates an $n \times n$ diagonal matrix whose diagonal entries are the components of the vector $x \in \mathbb{R}^n$
`eye(n)`	Creates an $n \times n$ identity matrix

Special Numbers in MATLAB

`pi`	The number $\pi = 3.1415\ldots$
`acos(a)`	The inverse cosine of the number a

Chapter 2: Solving Linear Equations

Editing and Number Commands

`format` Changes the numbers display format to standard five digit format

`format long`	Changes display format to 15 digits
`format rational`	Changes display format to rational numbers
`format short e`	Changes display to five digit floating point numbers

Vector Commands

`x.*y`	Componentwise multiplication of the vectors `x` and `y`
`x./y`	Componentwise division of the vectors `x` and `y`
`x.^y`	Componentwise exponentiation of the vectors `x` and `y`

Matrix Commands

`A([i j],:) = A([j i],:)`
Swaps i^{th} and j^{th} rows of matrix A

`A\b` Solves the system of linear equations associated with the augmented matrix $(A|b)$

`x = linspace(xmin,xmax,N)`
Generates a vector `x` whose entries are N equally spaced points from `xmin` to `xmax`

`x = xmin:xstep:xmax`
Generates a vector whose entries are equally spaced points from `xmin` to `xmax` with stepsize `xstep`

`[x,y] = meshgrid(XMIN:XSTEP:XMAX,YMIN:YSTEP:YMAX);`
Generates two vectors x and y. The entries of x are values from `XMIN` to `XMAX` in steps of `XSTEP`. Similarly for y.

`rand(m,n)` Generates an $m \times n$ matrix whose entries are randomly and uniformly chosen

	from the interval $[0, 1]$
`rref(A)`	Returns the reduced row echelon form of the $m \times n$ matrix A
	the matrix after each step in the row reduction process
`rank(A)`	Returns the rank of the $m \times n$ matrix A

Graphics Commands

`plot(x,y)`	Plots a graph connecting the points $(x(i), y(i))$ in sequence
`xlabel('labelx')`	Prints `labelx` along the x axis
`ylabel('labely')`	Prints `labely` along the y axis
`surf(x,y,z)`	Plots a three dimensional graph of $z(j)$ as a function of $x(j)$ and $y(j)$
`hold on`	Instructs MATLAB to *add* new graphics to the previous figure
`hold off`	Instructs MATLAB to *clear* figure when new graphics are generated
`grid`	Toggles grid lines on a figure
`axis('equal')`	Forces MATLAB to use equal x and y dimensions
`view([a b c])`	Sets viewpoint from which an observer sees the current 3-D plot
`zoom`	Zoom in and out on 2-D plot. On each mouse click, axes change by a factor of 2

Special Numbers and Functions in MATLAB

`exp(x)`	The number e^x where $e = \texttt{exp}(1) = 2.7182\ldots$
`sqrt(x)`	The number $\sqrt{x}$
`i`	The number $\sqrt{-1}$

Chapter 3: Matrices and Linearity

Matrix Commands

`A*x`	Performs the matrix vector product of the matrix A with the vector x
`A*B`	Performs the matrix product of the matrices A and B
`size(A)`	Determines the numbers of rows and columns of a matrix A
`inv(A)`	Computes the inverse of a matrix A

Program for Matrix Mappings

†`map`	Allows the graphic exploration of planar matrix mappings

Chapter 4: Solving Ordinary Differential Equations

Special Functions in MATLAB

`sin(x)`	The number $\sin(x)$
`cos(x)`	The number $\cos(x)$

Matrix Commands

`eig(A)`	Computes the eigenvalues of the matrix A
`null(A)`	Computes the solutions to the homogeneous equation $Ax = 0$

Programs for the Solution of ODEs

†`dfield8`	Displays graphs of solutions to differential equations
†`pline`	Dynamic illustration of phase line plots for single autonomous differential equations
†`pplane10`	Displays phase space and time series plots for systems of autonomous differential equations

Chapter 7: Determinants and Eigenvalues

Matrix Commands

`det(A)`	Computes the determinant of the matrix A
`poly(A)`	Returns the characteristic polynomial of the matrix A
`sum(v)`	Computes the sum of the components of the vector v
`trace(A)`	Computes the trace of the matrix A
`[V,D] = eig(A)`	Computes eigenvectors and eigenvalues of the matrix A

Chapter 8: Linear Maps and Changes of Coordinates

Vector Commands

†bcoord	Geometric illustration of planar coordinates by vector addition
†ccoord	Geometric illustration of coordinates relative to two bases

Chapter 10: Orthogonality

Matrix Commands

`orth(A)`	Computes an orthonormal basis for the column space of the matrix A
`[Q,R] = qr(A,0)`	Computes the QR decomposition of the matrix A

Graphics Commands

`axis([xmin,xmax,ymin,ymax])`
Forces MATLAB to use in a twodimensional plot the intervals `[xmin,xmax]` resp. `[ymin,ymax]` labeling the x- resp. y-axis

`plot(x,y,'o')` Same as `plot` but now the points $(x(i), y(i))$ are marked by circles and no longer connected in sequence

Chapter 11: Matrix Normal Forms

Vector Commands

`real(v)`	Returns the vector of the real parts of the components of the vector v
`imag(v)`	Returns the vector of the imaginary parts of the components of the vector v

Index