

Differences between v2.1 and v2.2 of the
NO BULLSHIT GUIDE TO LINEAR ALGEBRA

Ivan Savov

2020-11-08

No bullshit guide to linear algebra

by Ivan Savov

Copyright © Ivan Savov, 2014, 2017. — All rights reserved.

Published by Minireference Co.

Montréal, Québec, Canada

minireference.com | [@minireference](https://fb.me/noBSguide) | fb.me/noBSguide

For inquiries, contact the author at ivan@minireference.com

Library and Archives Canada Cataloguing in Publication

Savov, Ivan, 1982-, author

— No bullshit guide to linear algebra / Ivan Savov. — Second edition.

ISBN 978-0-9920010-2-5 (softcover)

1. Algebras, Linear—Textbooks. 2. Mathematics—Textbooks.

3. Textbooks. I. Title.

QA184.2.S29 2017

512'.5

C2017-902491-4

Cover image: Jim Hefferon and Nicholas Longo, CC-BY-SA-2.5

https://commons.wikimedia.org/wiki/File:Linalg_parallelepiped.png

Second edition

v2.1.2

git commits master@1058:7de6560 + vers/LA@1179:91e8d55

ISBN 978-0-9920010-2-5

10 9 8 7 6 5 4 3 2 1

5.3	Change of basis for matrices	174
5.4	Invertible matrix theorem	180
5.5	Linear transformations problems	187
6	Theoretical linear algebra	191
6.1	Eigenvalues and eigenvectors	192
6.2	Special types of matrices	206
6.3	Abstract vector spaces	213
6.4	Abstract inner product spaces	218
6.5	Gram–Schmidt orthogonalization	224
6.6	Matrix decompositions	228
6.7	Linear algebra with complex numbers	233
6.8	Theory problems	251
7	Applications	265
7.1	Balancing chemical equations	266
7.2	Input–output models in economics	267
7.3	Electric circuits	270
7.4	Graphs	275
7.5	Fibonacci sequence	278
7.6	Linear programming	281
7.7	Least squares approximate solutions	282
7.8	Computer graphics	293
7.9	Cryptography	304
7.10	Error-correcting codes	315
7.11	Fourier analysis	322
7.12	Applications problems	335
8	Probability theory	347
8.1	Probability distributions	347
8.2	Markov chains	358
8.3	Google’s PageRank algorithm	364
8.4	Probability problems	371
9	Quantum mechanics	375
9.1	Introduction	376
9.2	Polarizing lenses experiment	382
9.3	Dirac notation for vectors	387
9.4	Quantum information processing	392
9.5	Postulates of quantum mechanics	395
9.6	Polarizing lenses experiment revisited	411
9.7	Quantum <u>physics</u> <u>mechanics</u> is not that weird	414
9.8	Quantum mechanics applications	419
9.9	Quantum mechanics problems	434

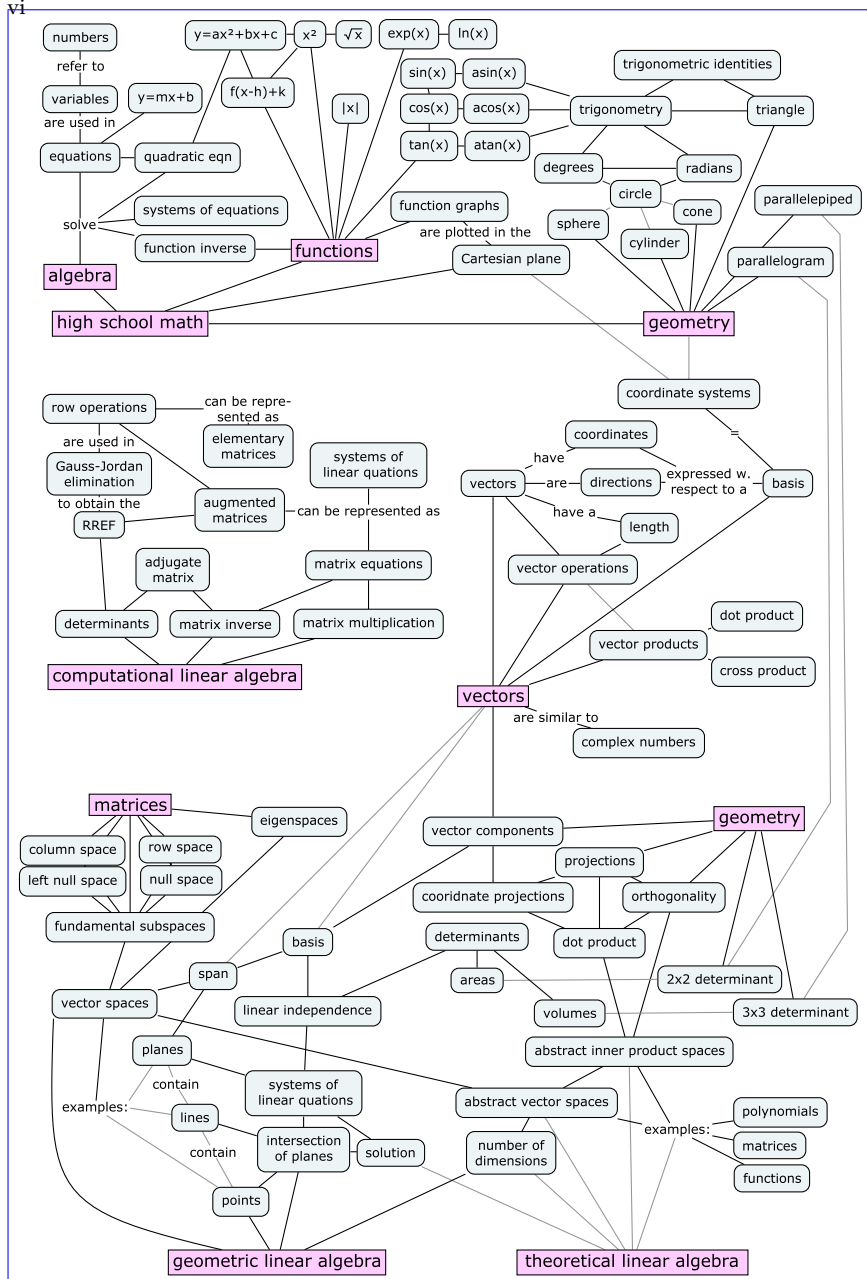


Figure 1: This concept map diagram shows all the topics and concepts and topics covered in this book and illustrates the connections between them. This book is all about linking the concepts together. Since it's a lot of stuff, we'll start slowly by reviewing prerequisite topics of high school math (Chapter 1), and gradually build your knowledge from there.

Consult the index on page 455 to find the exact location in the book where each concept is defined.

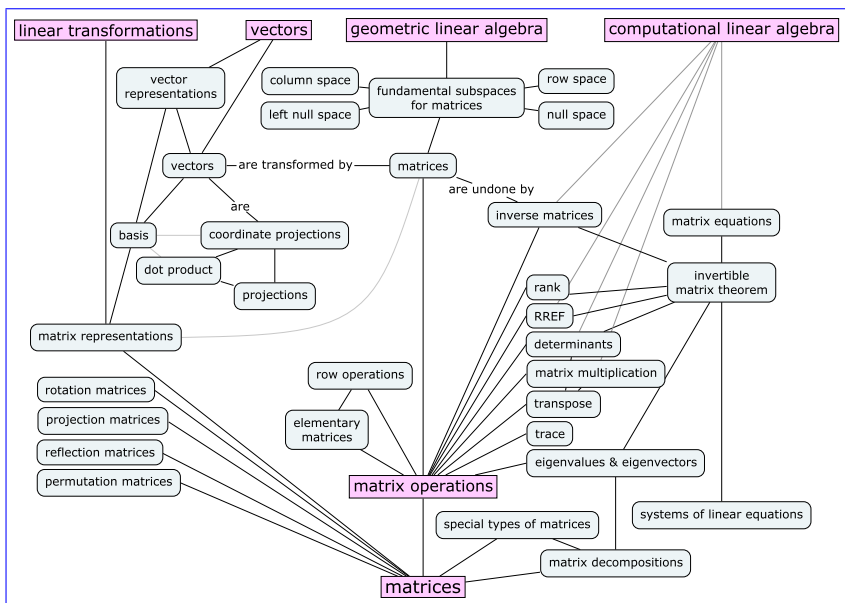


Figure 4: Matrix **operations** and **matrix** computations play an important role throughout this book. Matrices are used to represent linear transformations, systems of linear equations, and various geometric **computations** **operations**.

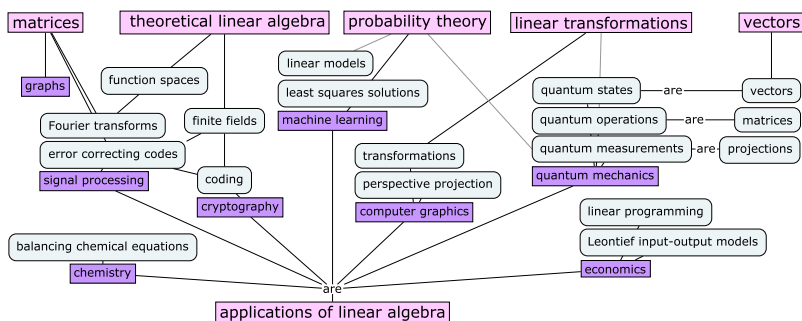


Figure 5: The book concludes with three chapters on linear algebra applications. In Chapter 7 we'll discuss applications to science, economics, business, computing, and signal processing. In Chapter 8 ~~on we'll explain~~ probability theory, and finally in Chapter 9 ~~on we'll introduce~~ quantum mechanics ~~serve as examples of advanced subjects that you can access once you learn linear algebra.~~

You can annotate the concept maps with your current knowledge of each concept to keep track of your progress. Add a single dot (•) next to all concepts you've heard of, two dots (••) next to concepts you think you know, and three dots (•••) next to concepts you've used in exercises and problems. By collecting some dots every week, you'll be able to move through the material in no time at all.

If you don't want to mark up your book, you can download a printable version of the concept maps here: bit.ly/LAcmaps.

The "point system" is a new idea with the goal of making readers more aware of how their knowledge is developing.

The idea is for readers to actively mark up concepts as unknown, known, and used and observe the progress: from a giant map with scary unknown terminology ...

... to a bunch of "completed levels" made up of concepts the reader knows well.

Looking for feedback about this idea from anyone who tries it out.

Did it feel motivational or just a chore?

Preface

This is a book about linear algebra and its applications. The material is presented at the level of a first-year university course, in an approachable style that cuts to the point. It covers both practical and theoretical aspects of linear algebra, with extra emphasis on explaining the connections between concepts and building a solid understanding of the material.

This book is designed to **give readers access to advanced math modelling tools** regardless of their academic background ~~regardless of their academic background~~. Since the book includes all the prerequisites needed to learn linear algebra, it's suitable for readers of any skill level—including those who don't feel comfortable with fundamental math concepts.

Why learn linear algebra?

Linear algebra is one of the most fundamental and all-around useful subjects in mathematics. The practical skills learned by studying linear algebra—such as manipulating vectors and matrices—form an essential foundation for applications in physics, computer science, statistics, machine learning, and many other fields of scientific study. Learning linear algebra can also be a lot of fun. Readers will experience numerous knowledge buzz ~~as they learn about moments when they feel math ideas “click” into place and understand~~ the connections between concepts, ~~and it's not uncommon to experience mind-expanding moments while studying this subject.~~

The ~~powerful concepts and~~ tools of linear algebra form a bridge to more advanced areas of mathematics. For example, learning about *abstract vector spaces* will help ~~students~~ readers recognize the common “vector space structure” in seemingly unrelated mathematical objects like matrices, polynomials, and functions. Linear algebra

techniques apply not only to standard vectors, but to *all* mathematical objects that are vector-like!

What's in this book?

Each section is a self-contained tutorial that covers the definitions, formulas, and explanations associated with a single topic. Check out the concept maps on the preceding pages to see the book's many topics and the connections between them.

The book begins with a review chapter on numbers, algebra, ~~sets~~, equations, functions, geometry, ~~trigonometry, vectors, and complex numbers and trigonometry~~ (Chapter 1). If you haven't previously studied these concepts, or if you feel your math ~~and vector~~ skills are a little "rusty," read this chapter and work through the exercises and problems provided. If you feel confident in your high school math abilities, jump straight to Chapter 2, where the linear algebra begins.

Chapters ~~3~~2–6 cover the core topics of linear algebra: vectors, bases, analytical geometry, matrices, linear transformations, matrix representations, vector spaces, inner product spaces, eigenvectors, and matrix decompositions. These chapters contain the material required for every university-level linear algebra course. Each section contains plenty of exercises so you can test your understanding as you read; and each chapter concludes with an extensive list of problems for further practice.

Chapters 7, 8, and 9 discuss various applications of linear algebra. Though this material isn't likely to appear on any ~~linear algebra~~ final exam, these chapters serve to demonstrate the power of linear algebra techniques and their relevance to many areas of science. The mini-course on quantum mechanics (Chapter 9) is unique to this book. Read this chapter to understand the fascinating laws of physics that govern the behaviour of atoms and photons.

Is this book for you?

The quick pace and lively explanations in this book provide interesting reading for students and non-students alike. Whether you're learning linear algebra for a course, reviewing material as a prerequisite for more advanced topics, or generally curious about the subject, this ~~guide book~~ will help you find your way in the land of linear algebra. ~~The tutorial format cuts quickly and clearly to the point—because we're all busy people with no time to waste!~~

Students and educators can use this book as the main textbook for any university-level linear algebra course. It contains everything students need to know to prepare for a linear algebra final exam. Don't

be fooled by the book's small size compared to other textbooks: it's all in here. The text is compact because we've distilled the essentials and removed the unnecessary crud.

Publisher

The ~~genesis of the no bullshit~~ starting point of the No Bullshit Guide textbook series dates back to my student days ~~7~~ when I was required to purchase expensive course textbooks, which were long and tedious to read. I said to myself, "Something must be done," and started a ~~textbook publishing~~ company to produce textbooks that explain math and physics concepts clearly, concisely, and affordably.

~~The goal of Minireference Publishing is to fix the first-year science textbook problem: mainstream textbooks are too expensive, boring, and limited in how they teach. We're creating a better alternative—one that's redefining readers' expectations about what textbooks should be. Print-on-demand and digital distribution strategies allow us to provide readers with high-quality textbooks at reasonable prices, making advanced math and science knowledge accessible to anyone interested in learning.~~

~~The secret behind the effectiveness of the no bullshit series is the spirit of continuous improvement. All Minireference authors are experts with years of teaching experience who co-own the books they write. Our authors maintain a direct connection with their readers by listening and responding to feedback. The combination of skilled authors and small editorial teams equipped with a modern publishing toolchain allows us to quickly respond to the feedback we receive, constantly improving our titles.~~ Minireference Co. is a publisher specializing in math and science. Our goal is to make advanced math modelling tools accessible to everyone by producing affordable textbooks that explain math concepts clearly and concisely. It's time for a break from traditional textbooks that are expensive, heavy, and tedious to read. The books in the No Bullshit Guide series are an efficient and enjoyable alternative for learning mathematical subjects.

About the author

I have been teaching math and physics for more than ~~15-17~~ years as a private tutor. Through ~~teaching this experience~~, I learned to ~~explain difficult concepts by breaking break~~ complicated ideas into smaller ~~chunks~~, interconnected chunks that are easy to understand. An interesting feedback loop occurs when students learn concepts in small, manageable chunks: they experience *knowledge buzz* when-

ever concepts “click” into place, and this excitement motivates them to continue learning more. I know this from first-hand experience, both as a teacher and as a student. I completed my undergraduate studies in electrical engineering, then stayed on to earn a M.Sc. in physics, and a Ph.D. in computer science from McGill University. ~~Nowadays I focus on teaching and writing effective lessons that help students and adult learners increase their math power.~~

Linear algebra played a central role throughout my studies. With this book, I want to share with you some of what I’ve learned about this expansive subject.

Ivan Savov
Montreal, ~~2017~~ 2020

Introduction

In recent years we've seen countless advances in science and technology. Modern science and engineering fields have developed advanced models for understanding the real world, predicting the outcomes of experiments, and building useful technology. Although we're still far from obtaining a "theory of everything" that can fully explain reality and predict the future, we do have a significant understanding of the natural world on many levels: physical, chemical, biological, ecological, psychological, and social. And, since mathematical models are leveraged throughout these fields of study, anyone interested in contributing to scientific and technological advances must also understand mathematics.

The linear algebra techniques you'll learn in this book are some of the most powerful mathematical modelling tools that exist. At the core of linear algebra lies a very simple idea: *linearity*. A function f is *linear* if it obeys the equation

$$f(ax_1 + bx_2) = af(x_1) + bf(x_2),$$

where x_1 and x_2 are any two inputs of the function. We use the term *linear combination* to describe any expression constructed from a set of variables by multiplying each variable by a constant and adding the results. In the above equation, the linear combination $ax_1 + bx_2$ of the inputs x_1 and x_2 is transformed into the linear combination $af(x_1) + bf(x_2)$ of the outputs of the function $f(x_1)$ and $f(x_2)$. **Essentially, linear functions transform a linear combinations-combination of inputs into the same linear combinations-combination of outputs.** If the input to the linear function f consists of five parts x_1 and three parts x_2 , then the output of the function will consist of five parts $f(x_1)$ and three parts $f(x_2)$. That's it, that's all! Now you know everything there is to know about linear algebra. The rest of the book is just details.

Linear models are super useful

A significant proportion of the ~~models used by scientists and engineers~~ math models used in science describe *linear relationships* between quantities. ~~Scientists, engineers, statisticians, business folk, and politicians~~ Mathematicians, scientists, engineers, and business analysts develop and use linear models to make sense of the systems they study. ~~In fact, linear models are often used to model even nonlinear (more complicated) phenomena.~~

~~There are several excellent reasons for using linear models. The first reason is that linear models are very good at approximating the real world. Linear models that represent nonlinear phenomena are referred to as linear approximations.~~

~~The second excellent reason to use linear algebra is that we can describe nonlinear phenomena by combining linear models with nonlinear transformations of the models' inputs or outputs. These techniques are often employed in machine learning: kernel methods are arbitrary, nonlinear transformations of the inputs of a linear model, and the sigmoid activation curve is used to transform the smoothly-varying output of a linear model into a hard yes or no decision, an on or off command, a 0 or 1 value, etc.~~

~~Perhaps the main reason linear models are widely used is~~ Linear models are popular because they are ~~easy to describe mathematically, and easy to "fit" to real-world systems~~ easy to describe mathematically. We can obtain the parameters of a linear model for a real-world system by analyzing the system's behaviour for relatively few inputs. Let's illustrate this important point with an example.

Example You ~~enter~~ re visiting an art gallery. Inside, the screen of a tablet computer is being projected onto a giant wall. Anything you draw on the tablet instantly appears projected onto the wall. However, the tablet's user interface doesn't give any indication about how to hold the tablet "right side up." How can you find the correct orientation of the tablet so your drawing won't appear rotated or upside-down?

~~This situation is directly analogous to the tasks scientists face every day when trying to model real-world systems.~~

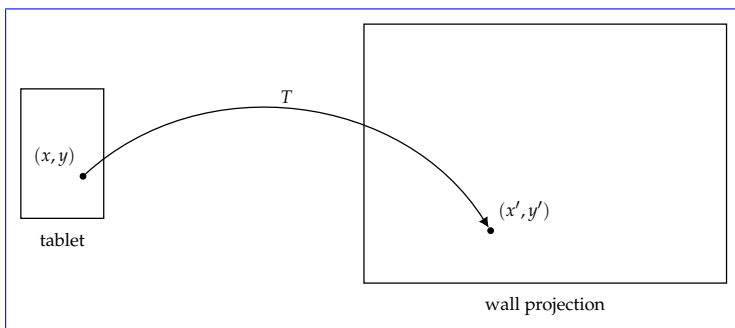


Figure 6: An unknown linear transformation T maps “tablet coordinates” to “wall coordinates.” How can we characterize T ?

The tablet’s screen is a two-dimensional *input space*, and the wall projection is a two-dimensional *output space*. We’re looking for the unknown transformation T that maps the pixels of the tablet screen (the input space) to the projection on the wall (the output space):

$$(x, y) \xrightarrow{T} (x', y').$$

This task is directly analogous to the tasks scientists and engineers face every day when trying to model real-world systems by observing how systems transform inputs to outputs. If the unknown transformation T is a linear transformation, we can learn its parameters very quickly.

An unknown linear transformation T maps “tablet coordinates” to “wall coordinates.” How can we characterize T ?

Let’s describe each pixel in the input space with a pair of coordinates (x, y) and each point on the wall with another pair of coordinates (x', y') . The unknown transformation T describes the mapping of tablet coordinates to wall coordinates:

$$(x, y) \xrightarrow{T} (x', y').$$

To uncover linear, you can learn what it is very quickly, using only two swipes on the tablet screen.

To understand how T transforms screen coordinates (x, y) to wall coordinates (x', y') , you can use the following three-step procedure. First, put a dot in the lower-left corner of the tablet to represent the *origin* $(0, 0)$ of the xy -coordinate system. Observe the location where the dot appears

on the wall—we'll call this location the origin of the $x'y'$ -coordinate system. Next, draw a short, this two-step "probing" procedure:

1. Draw a horizontal line on the tablet to represent the x -direction $(1,0)$, and observe the transformed $T(1,0)$ that appears in the input space $\rightarrow = (1,0)$. You observe the output ↗ projected on the wall. ~~Last, draw~~ This tells you horizontal lines are transformed to northeast diagonal lines in the wall-projection space.
2. Draw a vertical line in the y -direction $(0,1) \uparrow = (0,1)$ on the tablet, and see the transformed $T(0,1)$ that appears on the wall. You observe the output ↖ appears on the wall. By noting how the xy -coordinate system is mapped to the $x'y'$ -coordinate system, you can determine the orientation in which you must hold the tablet so your drawing appears upright when projected.
~~Knowing the outputs of a linear transformation T for all "directions" in its input space is a complete characterization of T .~~ This means vertical lines on the tablet screen turn into northwest diagonal lines when projected on the wall.

In the case of the tablet and the wall, we're looking for an unknown transformation T from a two-dimensional input space to a two-dimensional output space. Since T is a linear transformation, it's possible to completely describe T with only two lines (one line for each dimension).

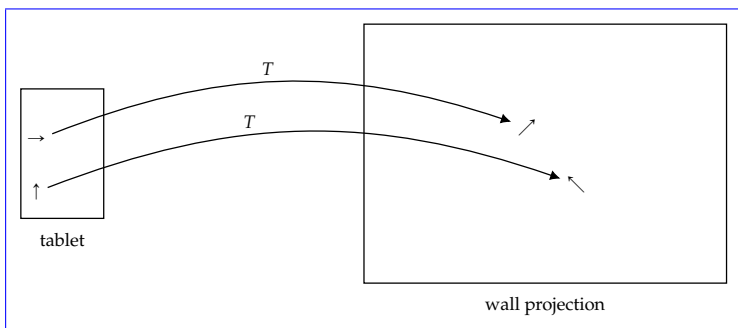


Figure 7: Drawing a short horizontal arrow \rightarrow on the tablet screen results in a northeast diagonal projection on the wall \nearrow . Drawing a vertical arrow \uparrow on the tablet results in a northwest diagonal line projected on the wall \nwarrow .

Here comes the interesting part: now that you know the outputs ↗ and ↖ produced for the two input directions, you can predict the linear transformation's output for any other input. Let's look at the math to see equations that show why this is true. ~~Can you~~

Suppose you want to predict what will appear on the wall if you draw an angled line on the tablet in the $(2, 3)$ -direction? First, locate the point $(2, 3)$ in the input space by moving 2-direction $(3, 2)$. The coordinates $(3, 2)$ describe a swipe with length 3 units in the x -direction and 2 units in the y -direction: $(2, 3) = (2, 0) + (0, 3) = 2(1, 0) + 3(0, 1)$. Then, using the fact that T is a linear transformation, we can predict the output of the transformation when the input is $(2, 3)$:

$$T(2, 3) = T(2(1, 0) + 3(0, 1)) = 2T(1, 0) + 3T(0, 1).$$

The projection of the diagonal line in the $(2, 3)$ -direction linear, the wall projection of this input will have a length equal to 2 times the unit 3 times the x -direction output $T(1, 0)$ plus 3 times the unit plus 2 times the y -direction output $T(0, 1)$. Knowing the outputs of the two lines $T(1, 0)$ and $T(0, 1)$ is sufficient:

$$T(3 \rightarrow + 2 \uparrow) = 3T(\rightarrow) + 2T(\uparrow) = 3 \nearrow + 2 \nwarrow.$$

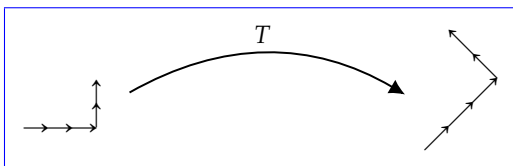


Figure 8: The linear transformation T maps the input $3 \rightarrow + 2 \uparrow$ to the output $3T(\rightarrow) + 2T(\uparrow) = 3 \nearrow + 2 \nwarrow$.

Knowing that the input \rightarrow produces the output \nearrow and the input \uparrow produces the output \nwarrow allows you to determine the linear transformation's output for any input (a, b) . Any all other inputs. Every input (a, b) can be expressed written as a linear combination: $(a, b) = a(1, 0) + b(0, 1)$. The $(a, b) = a(1, 0) + b(0, 1) = a \rightarrow + b \uparrow$. Since you know T is linear, you know the corresponding output will be $T(a, b) = aT(1, 0) + bT(0, 1)$. Since we know $T(1, 0)$ and $T(0, 1)$, we can calculate $T(a, b)$.

Don't worry if you can't follow all the math in this example. It's the concepts that are essential right now, and we'll have plenty of time to work on the math behind the concepts in the rest of the book

$$T(a \rightarrow + b \uparrow) = aT(\rightarrow) + bT(\uparrow) = a \nearrow + b \nwarrow.$$

Since you can predict the output of T for all possible inputs, you have obtained a complete characterization of the linear transformation T .

The probing procedure we used to characterize the two-dimensional tablet-to-wall linear transformation (denoted $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$) can be used to study arbitrary linear transformations with n -dimensional inputs and m -dimensional outputs (denoted $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$). Knowing the outputs of a linear transformation T for all “directions” in its input space gives us a complete characterization of T .

TL;DR ~~Linearity~~ The linear property allows us to analyze multidimensional ~~processes and transformations~~ systems and processes by studying their effects on a small set of inputs. This is the essential reason linear models are ~~so prominent~~ used so widely in science. ~~Probing a linear system with each “input direction” is enough to completely characterize the system.~~ Without this linear structure, characterizing the behaviour of unknown input-output systems is ~~would be~~ a much harder task. ~~Linear algebra is the study of linear structure, in all its details. The theoretical results and computational procedures you’ll learn apply to all things linear and vector-like.~~

Linear transformations

Linear transformations will be a central topic throughout this book. You can think of linear transformations as “vector functions” and understand their properties as analogous to the properties of the regular functions you’re familiar with. The action of a function on a number is similar to the action of a linear transformation on a vector:

function $f : \mathbb{R} \rightarrow \mathbb{R}$ \Leftrightarrow linear transformation $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$

input $x \in \mathbb{R}$ \Leftrightarrow input $\vec{x} \in \mathbb{R}^n$

output $f(x) \in \mathbb{R}$ \Leftrightarrow output $T(\vec{x}) \in \mathbb{R}^m$

inverse function f^{-1} \Leftrightarrow inverse transformation T^{-1}

~~zeros of~~ roots of f \Leftrightarrow kernel of T

Studying linear algebra will expose you to many new topics associated with linear transformations. You’ll learn about concepts like vector spaces, projections, rotations, and orthogonalization procedures. Indeed, a first linear algebra course introduces many advanced, abstract ideas; yet all the new ideas you’ll encounter can be seen as extensions of ideas you’re already familiar with. Linear algebra is the vector-upgrade to your high school knowledge of functions.

Prerequisites

To understand linear algebra, you must have some preliminary knowledge of fundamental math concepts like numbers, equations, and functions. For example, you should be able to tell me the meaning of the parameters m and b in the equation $f(x) = mx + b$. If you do not feel confident about your basic math skills, don't worry. Chapter 1 is a prerequisites chapter specially designed to help bring you quickly up to speed on the ~~material of~~ high school math. ~~It also contains a short summary of vectors concepts usually taught in the first week of Physics 101, and a section on complex numbers (Section ??). You should read about complex numbers at some point because we'll use complex numbers in Section 6.7 later in the book~~material.

Executive summary

The book is organized into nine chapters. Chapters 2 ~~through~~ 6 are the core of linear algebra. Chapters 7 through 9 contain optional reading about linear algebra applications. The concept maps on pages vi, viii, and ix illustrate the connections between the topics we'll cover. I know the maps ~~are teeming with concepts~~may seem informationally intimidating at first sight, but don't worry—the book is split into tiny chunks, and we'll navigate the material step by step. It will be like Mario World, but in n dimensions and with a lot of bonus levels.

Chapter 2 is ~~an introduction to the subject of~~ a general introduction to linear algebra. Linear algebra is the math of vectors and matrices, so we'll start by defining the mathematical operations we can perform on vectors and matrices.

In Chapter 3, we'll tackle the computational aspects of linear algebra. By the end of this ~~course, chapter~~ you'll know how to solve systems of equations, transform a matrix into its ~~reduced row echelon form~~ reduced row echelon form, compute the product of two matrices, and find the ~~determinant and the inverse~~ determinant and the inverse of a square matrix. Each of these computational tasks can be tedious to carry out by hand and can require lots of steps. There is no way around this; we must do the grunt work before we get to the cool stuff.

In Chapter 4, we'll review the properties and the equations that describe basic geometric objects like points, lines, and planes. We'll learn how to compute projections onto vectors, projections onto planes, and distances between objects. We'll also review the meaning of vector coordinates, which are lengths measured with respect

to a basis. We'll learn about linear combinations of vectors, the span of a set of vectors, and formally define what a vector space is. In Section 4.5, we'll learn how to use the reduced row echelon form of a matrix, to describe the fundamental spaces associated with the matrix. vector space is.

Chapter 5 is about linear transformations. Armed with the computational tools from Chapter 3 and the geometric intuition from Chapter 4, we can tackle the core subject of linear algebra: linear transformations. We'll explore in detail the correspondence between linear transformations ($T : \mathbb{R}^n \rightarrow \mathbb{R}^m$) and their representation as $m \times n$ matrices. We'll also learn how the coefficients entries in a matrix representation depend on the choice of basis for the input and output spaces of the transformation. Section 5.4 on the invertible matrix theorem serves as a midway checkpoint for your understanding of linear algebra. This theorem connects several seemingly disparate concepts: reduced row echelon forms, matrix inverses, row spaces, column spaces, and determinants. The invertible matrix theorem links all these concepts and highlights the properties of invertible linear transformations that distinguish them from non-invertible transformations. Invertible transformations are one-to-one correspondences (bijections) between vectors in the input space and vectors in the output space.

Chapter 6 covers more advanced theoretical topics of linear algebra. We'll define the eigenvalues and the eigenvectors eigenvalues and the eigenvectors of a square matrix. We'll see how the eigenvalues of a matrix tell us important information about the properties of the matrix. We'll learn about some, and learn about the special names given to different types of matrices, based on the properties of their eigenvalues. In Section 6.3 we'll learn about abstract vector spaces discuss abstract vector spaces. Abstract vectors are mathematical objects that—like vectors—have components and can be scaled, added, and subtracted by manipulating their components. Section 6.7 will discuss linear algebra with complex numbers. Instead of working with vectors with real coefficients, we'll see how to do linear algebra with vectors that have complex coefficients. This section serves as a review of all the material in the book. We'll revisit all the key concepts and find out how they are affected when working with complex numbers.

In Chapter 7, we'll discuss the applications of linear algebra. If you've done your job learning the material in the first seven six chapters, you'll get to learn all the cool things you can do with linear algebra. Chapter 8 will introduce the basic concepts of probability theory. Chapter 9 contains an introduction to quantum mechanics.

The sections in the book are self-contained so you can read them

in any order. Feel free to skip ahead to the parts that you want to learn first. That being said, the material is ordered to provide an optimal knowing-what-you-need-to-know-before-learning-what-you-want-to-know experience

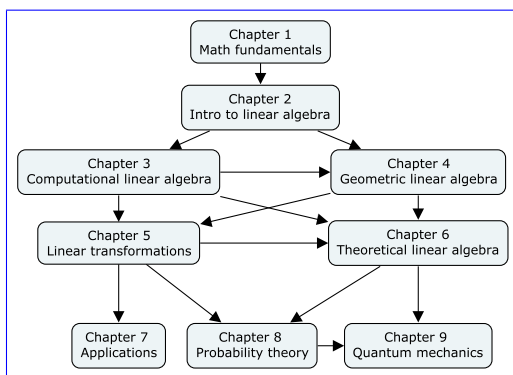


Figure 9: The dependency structure of the chapters in the book.

Figure 9 shows the prerequisite structure between the chapters. If you're new to linear algebra, it would be best to read **everything the chapters** in order. If you **find yourself stuck on a concept at some point**, refer to the concept maps to see if **already have some experience with the subject**, you can skip ahead to the parts you're missing some prerequisites and flip to the section of the book that will help you fill in your knowledge gap accordingly **interested in**. Use the table of contents, the concept maps on pages vi–ix, and the index on page 455 to navigate the book. The chapters and sections in the book are self-contained so they will make sense even if you read them out of order.

Difficulty level

In terms of difficulty, I must prepare you to get ready for some serious uphill pushes. As your personal “trail guide” up the mountain of linear algebra, it's my obligation to warn you about the difficulties that lie ahead, **so that you can mentally prepare for a good challenge**.

Linear algebra is a difficult subject because it requires developing your computational skills, your geometric intuition, and your abstract **thinking—thinking—all at the same time!** The computational aspects of linear algebra are not particularly difficult, but they can be boring and repetitive. You'll have to carry out hundreds of steps of basic arithmetic. The geometric problems you'll encounter in Chapter 4 can be tough at first, but they'll get easier once you learn to draw

Chapter 1

Math fundamentals

In this chapter we'll review the fundamental ideas of **mathematics** ~~which are the~~ mathematics—the prerequisites for learning linear algebra. We'll define the different types of numbers and the concept of a function, which is a transformation that takes numbers as inputs and produces numbers as outputs. Linear algebra is the extension of these ideas to many dimensions: instead of "doing math" doing math with numbers and functions, in linear algebra we'll be "doing math" doing math with vectors and linear transformations.

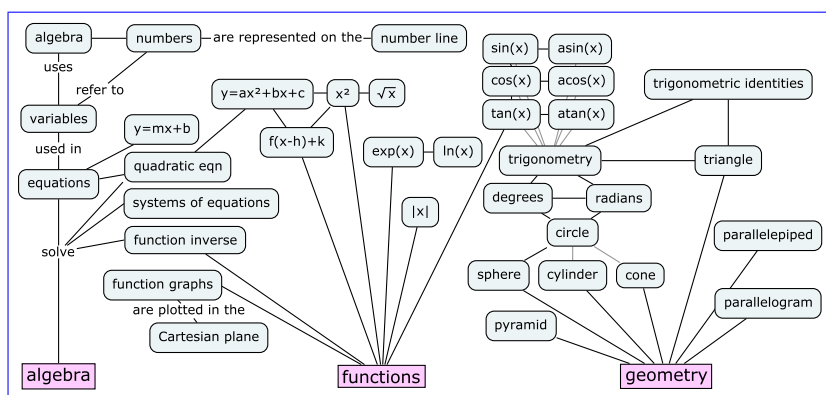


Figure 1.1: A concept map showing the mathematical topics covered in this chapter. We'll learn how to solve equations using algebra, how to model the world using functions, and some important facts about geometry. The material in this chapter is required for your understanding of the more advanced topics in this book.

Chapter 2

Intro to linear algebra

The first chapter reviewed core ideas of mathematics. Now that we're done with the prerequisites, we can begin the main discussion of linear algebra: the study of vectors and matrices.

2.1 Definitions

Vectors and matrices are the objects of study in linear algebra, and in this chapter we'll define them and learn the basic operations we can perform on them.

We denote the set of n -dimensional vectors with real coefficients as \mathbb{R}^n . A vector $\vec{v} \in \mathbb{R}^n$ is an n -tuple of real numbers.¹ For example, a three-dimensional vector is defined as a triple of the form numbers:

$$\vec{v} \equiv_{\text{def}} (v_1, v_2, v_3) \in (\mathbb{R}, \mathbb{R}, \mathbb{R}) \equiv \mathbb{R}^3.$$

To specify the vector \vec{v} , we must specify the values for its three components: v_1 , v_2 , and v_3 . We'll use the terms components and coordinates interchangeably throughout the book.

A matrix $A \in \mathbb{R}^{m \times n}$ is a rectangular array of real numbers with m rows and n columns. For example, a 3×2 matrix looks is defined like this:

$$A \equiv_{\text{def}} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \in \mathbb{R}^{3 \times 2}.$$

To specify the matrix A , we need to specify the values of its six components, entries: a_{11} , a_{12} , \dots , a_{21} , a_{22} , a_{31} , and a_{32} .

¹The notation " $s \in S$ " is read " s is an element of S " or " s in S ."

E2.2 Given the matrices $A = \begin{bmatrix} 1 & 3 \\ 4 & 5 \end{bmatrix}$ and $B = \begin{bmatrix} -1 & 0 \\ 3 & 3 \end{bmatrix}$, and the vectors $\vec{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\vec{w} = \begin{bmatrix} -3 \\ -4 \end{bmatrix}$, compute the following expressions.

- a) $A\vec{v}$ b) $B\vec{v}$ c) $A(B\vec{v})$ d) $B(A\vec{v})$ e) $A\vec{w}$
 f) $B\vec{w}$

2.2 a) $A\vec{v} = \begin{bmatrix} 7 \\ 14 \end{bmatrix}$; b) $B\vec{v} = \begin{bmatrix} -1 \\ 9 \end{bmatrix}$; c) $A(B\vec{v}) = \begin{bmatrix} 26 \\ 41 \end{bmatrix}$; d) $B(A\vec{v}) = \begin{bmatrix} -7 \\ 63 \end{bmatrix}$;
 e) $A\vec{w} = \begin{bmatrix} -15 \\ -32 \end{bmatrix}$; f) $B\vec{w} = \begin{bmatrix} 3 \\ -21 \end{bmatrix}$.

E2.3 Find the coefficients-components v_1 and v_2 of the vector $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ so that $E\vec{v} = 3\vec{e}_2 - 2\vec{e}_1$, where E is the following matrix:

$$E = \begin{bmatrix} | & | \\ \vec{e}_1 & \vec{e}_2 \\ | & | \end{bmatrix}.$$

2.3 $v_1 = -2, v_2 = 3$.

What next?

We won't bring geometry, vector spaces, algorithms, and the applications of linear algebra into the mix all at once. Instead, let's start with the basics. Since linear algebra is about vectors and matrices, let's define vectors and matrices precisely, and describe the math operations we can perform on them.

2.2 Vector operations

Section ?? introduced some basic notions about vectors. Understanding vectors is so important for linear algebra that it's worth going beyond the rudimentary understanding of vectors as "directional quantities," and so we took the time to describe vectors more abstractly—as math objects. With vectors defined, our next step is to specify their properties and the operations we can perform on vectors. This is what this section is all about.

OperationsDefinitions

Consider the vectors $\vec{u} = (u_1, u_2, u_3)$ and $\vec{v} = (v_1, v_2, v_3)$, and an arbitrary constant $\alpha \in \mathbb{R}$. Vector algebra can be summarized as the following operations:

- Addition: $\vec{u} + \vec{v} = (\underbrace{u_1 + v_1, u_2 + v_2, u_3 + v_3}_{\text{components}}) \stackrel{\text{def}}{=} (\underbrace{u_1 + v_1, u_2 + v_2, u_3 + v_3}_{\text{coefficients}})$

Scaling by a constant

Scaling is an operation that takes a number and a vector as inputs and produces a vector output:

$$\text{scalar-mult} : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n.$$

There is no symbol to denote scalar multiplication—we just write the **scaling-scale** factor in front of the vector and the multiplication is implicit.

Multiplying the vector \vec{u} by the **scaling-scale** factor α is equivalent to multiplying each component of the vector by α :

$$\vec{w} = \alpha \vec{u} \quad \Leftrightarrow \quad w_i = \alpha u_i.$$

For example, choosing $\alpha = 2$, we obtain the vector $\vec{w} = 2\vec{u}$, which is two times longer than the vector \vec{u} :

$$\vec{w} = (w_1, w_2, w_3) = (2u_1, 2u_2, 2u_3) = 2(u_1, u_2, u_3) = 2\vec{u}.$$

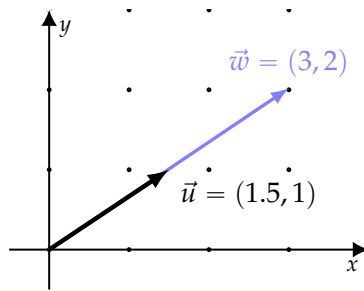


Figure 2.1: Vectors \vec{u} and \vec{w} are related by the equation $\vec{w} = 2\vec{u}$.

Vector products

We'll now define the *dot product* and the *cross product*: two geometric operations useful for working with three-dimensional vectors.

Dot product

The *dot product* takes two vectors as inputs and produces a single, real number as an output:

$$\cdot : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}.$$

Division by the length squared transforms the two appearances of the vector \vec{d} into the unit vectors \hat{d} needed for the projection formula:

$$\Pi_{\hat{d}}(\vec{v}) = \underbrace{(\vec{v} \cdot \hat{d})}_{\|\vec{v}\| \cos \theta} \hat{d} = \left(\vec{v} \cdot \frac{\vec{d}}{\|\vec{d}\|} \right) \frac{\vec{d}}{\|\vec{d}\|} = \left(\frac{\vec{v} \cdot \vec{d}}{\|\vec{d}\|^2} \right) \vec{d} = \Pi_{\vec{d}}(\vec{v}).$$

Remember these projection formulas well because we'll need them later: when computing projections onto planes (Section 4.2), when computing [vector](#) coordinates (Section 4.3), and when describing [the](#) change-of-basis [operations](#) [operation](#) (Section 5.3).

Discussion

This section elaborated on the properties of n -dimensional vectors, which are ordered tuples (lists) of n [coefficients](#) [components](#). It's important to think of vectors as whole mathematical objects, rather than as [coefficients](#) [components](#). Although vector operations boil down to manipulations of their [coefficients](#) [components](#), vectors are most useful (and best understood) when you think of them as whole objects that have components, rather than focussing on their components.

Links

[Nice illustration of the cross product]

<http://lucasvb.tumblr.com/post/76812811092/>

[Vectors explained by 3Blue1Brown]

https://youtube.com/watch?v=fNk_zzaMoSs

[Cross products explained by 3Blue1Brown]

<https://youtube.com/watch?v=eu6i7WJJeinw>

Exercises

E2.4 Given the vectors $\vec{u} = (1, 1, 0)$ and $\vec{v} = (0, 0, 3)$, compute the following vector expressions:

a) $\vec{u} + \vec{v}$

b) $\vec{u} - \vec{v}$

c) $3\vec{u} + \vec{v}$

d) $\|\vec{u}\|$

2.4 **a)** $(1, 1, 3)$; **b)** $(1, 1, -3)$; **c)** $(3, 3, 3)$; **d)** $\sqrt{2}$.

E2.5 Given $\vec{v} = (1, 2, 3)$ and $\vec{w} = (0, 1, 1)$, compute the following vector products: **a)** $\vec{v} \cdot \vec{w}$; **b)** $\vec{v} \times \vec{w}$; **c)** $\vec{w} \times \vec{v}$; **d)** $\vec{w} \times \vec{w}$.

2.5 **a)** 5; **b)** $(-1, -1, 1)$; **c)** $(1, 1, -1)$; **d)** $(0, 0, 0)$.

Multiplication by a constant

Recall that *scaling* is another word for multiplication by a constant. Given a number α and a matrix A , we can *scale* A by α as follows:

$$\alpha A = \alpha \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} \\ \alpha a_{21} & \alpha a_{22} \\ \alpha a_{31} & \alpha a_{32} \end{bmatrix}.$$

Matrix-vector multiplication

The result of the matrix-vector product between a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $\vec{v} \in \mathbb{R}^n$ is an m -dimensional vector:

$$\text{matrix-vector product} : \mathbb{R}^{m \times n} \times \mathbb{R}^n \rightarrow \mathbb{R}^m.$$

The formula for the matrix-vector product is

$$\vec{w} = A\vec{v} \quad \Leftrightarrow \quad w_i = \sum_{j=1}^n a_{ij}v_j, \quad \forall i \in [1, \dots, m].$$

For example, the product of a 3×2 matrix A and the 2×1 column vector \vec{v} results in a 3×1 vector:

$$\begin{aligned} A\vec{v} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = v_1 \underbrace{\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}}_{\text{column picture}} + v_2 \underbrace{\begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}}_{\text{column picture}} \\ &= \left. \begin{bmatrix} (a_{11}, a_{12}) \cdot \vec{v} \\ (a_{21}, a_{22}) \cdot \vec{v} \\ (a_{31}, a_{32}) \cdot \vec{v} \end{bmatrix} \right\} \text{row picture} \\ &= \begin{bmatrix} a_{11}v_1 + a_{12}v_2 \\ a_{21}v_1 + a_{22}v_2 \\ a_{31}v_1 + a_{32}v_2 \end{bmatrix} \in \mathbb{R}^{3 \times 1}. \end{aligned}$$

Note the two equivalent ways to understand the matrix-vector product: the column picture and the row picture. In the column picture, the multiplication of the matrix A by the vector \vec{v} produces a **linear combination of the columns of the matrix**. In the row picture, multiplication of the matrix A by the vector \vec{v} produces a column vector with coefficients components equal to the **dot products of the rows of the matrix A with the vector \vec{v}** .

Matrix-matrix multiplication

The matrix product AB of matrices $A \in \mathbb{R}^{m \times \ell}$ and $B \in \mathbb{R}^{\ell \times n}$ results in an $m \times n$ matrix:

$$\text{matrix-product} : \mathbb{R}^{m \times \ell} \times \mathbb{R}^{\ell \times n} \rightarrow \mathbb{R}^{m \times n}.$$

The formula for ~~the matrix product~~ matrix multiplication computes the dot product between each row of A and each column of B :

$$C = AB \quad \Leftrightarrow \quad c_{ij} = \sum_{k=1}^{\ell} a_{ik} b_{kj}, \forall i \in [1, \dots, m], j \in [1, \dots, n].$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} \end{bmatrix} \in \mathbb{R}^{3 \times 2}.$$

Transpose

The transpose matrix A^T is defined by the formula $a_{ij}^T = a_{ji}$. We obtain the transpose by “flipping” the matrix through its diagonal:

$$^T : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{n \times m}$$

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{bmatrix}^T = \begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \\ \alpha_3 & \beta_3 \end{bmatrix}.$$

Note that entries on the diagonal of the matrix do not change when we apply the transpose operation.

Properties of the transpose operation

- $(A + B)^T = A^T + B^T$
- $(AB)^T = B^T A^T$
- $(ABC)^T = C^T B^T A^T$
- $(A^T)^{-1} = (A^{-1})^T$

Vectors as matrices

A vector is a special type of matrix. You can treat a vector $\vec{v} \in \mathbb{R}^n$ either as a *column vector* ($n \times 1$ matrix) or as a *row vector* ($1 \times n$ matrix).

Inner product

Recall the definition of the *dot product* or *inner product* for vectors:

$$\cdot : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \quad \Leftrightarrow \quad \vec{u} \cdot \vec{v} \equiv \sum_{i=1}^n u_i v_i.$$

If we think of vectors as *column* vectors, we can write the dot product in terms of the matrix transpose operation T and the standard rules of matrix multiplication:

$$\vec{u} \cdot \vec{v} \equiv \vec{u}^T \vec{v} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = u_1 v_1 + u_2 v_2 + u_3 v_3.$$

The dot product for vectors is thus a special case of matrix multiplication. Alternatively, we could say that matrix multiplication is defined in terms of the dot product.

Outer product

Consider again two *column* vectors \vec{u} and \vec{v} ($n \times 1$ matrices). We obtain the inner product by applying the transpose to the *first* vector in the product: $\vec{u}^T \vec{v} \equiv \vec{u} \cdot \vec{v}$. Instead, if we apply the transpose to the *second* vector, we'll obtain the **outer product** of \vec{u} and \vec{v} . The outer product operation takes pairs of vectors as inputs and produces matrices as outputs:

$$\text{outer-product} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}.$$

For example, the outer product of two vectors in \mathbb{R}^3 is

$$\vec{u} \vec{v}^T = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{bmatrix} \in \mathbb{R}^{3 \times 3}.$$

Observe that the matrix-matrix product **between** of a 3×1 matrix and a 1×3 matrix results in a 3×3 matrix.

In Section 4.2 we'll see how the outer product is used to build *projection matrices*. For example, the matrix that corresponds to the projection onto the x -axis is $M_x \equiv \hat{i} \hat{i}^T \in \mathbb{R}^{3 \times 3}$. The x -projection of any vector \vec{v} is computed as the matrix-vector product, $M_x \vec{v} = \hat{i} \hat{i}^T \vec{v} = \hat{i} (\hat{i} \cdot \vec{v}) = v_x \hat{i}$. More on that later.

value $100mx$. This input-output proportionality does not hold for nonlinear terms.

In this section we'll discuss the special properties of expressions and equations containing only linear terms.

Introduction

A single-variable function takes as its input a real number x and outputs a real number y . The $f(x)$. The type signature of this class of functions function is

$$f: \mathbb{R} \rightarrow \mathbb{R}.$$

The most general linear function from \mathbb{R} to \mathbb{R} looks like this:

$$\underline{y} \equiv f(x) = mx,$$

where $m \in \mathbb{R}$ is called the *coefficient* of x . The action of a linear function is to multiply the input by the constant m . So far, so good.

Example of composition of linear functions Given the linear functions $f(x) = 2x$ and $\underline{g(y) = 3y}$ $g(x) = 3x$, what is the equation of the function $\underline{h(x) = g \circ f(x) = g(f(x))}$ $h(x) \stackrel{\text{def}}{=} g \circ f(x) = g(f(x))$? The composition of $f(x) = 2x$ and $\underline{g(y) = 3y}$ $g(x) = 3x$ is the function $h(x) = g(f(x)) = 3(2x) = 6x$. Note the composition of two linear functions is also a linear function. The coefficient of h is equal to the product of the coefficients of f and g .

Definition

Linear functions map any linear combination of inputs to the same linear combination of outputs. A function f is *linear* if it satisfies the equation

$$f(\alpha x_1 + \beta x_2) = \alpha f(x_1) + \beta f(x_2),$$

for any two inputs x_1 and x_2 , and for all constants α and β .

Lines are not linear functions!

Consider the equation of a line:

$$l(x) = mx + b,$$

where the constant m corresponds to the slope of the line, and the constant $\underline{b = f(0)}$ is the y -intercept of the line $b = f(0)$ is its initial value. A line $l(x) = mx + b$ with $b \neq 0$ is *not* a linear function. This

Theoretical linear algebra

Linear algebra will teach you how to reason about vectors and matrices in an abstract way. By thinking abstractly, you'll be able to extend your geometric intuition of two and three-dimensional problems to problems in higher dimensions. Much *knowledge buzz* awaits as you learn about new mathematical ideas and develop new ways of thinking.

You're no doubt familiar with the normal coordinate system made of two orthogonal axes: the x -axis and the y -axis. A vector $\vec{v} \in \mathbb{R}^2$ is specified in terms of its coordinates (v_x, v_y) with respect to these axes. When we say $\vec{v} = (v_x, v_y)$, what we really mean is $\vec{v} = v_x \hat{i} + v_y \hat{j}$, where \hat{i} and \hat{j} are unit vectors that point along the x - and y -axes. As it turns out, we can use many other kinds of coordinate systems to represent vectors. A *basis* for \mathbb{R}^2 is any set of two vectors $\{\hat{e}_1, \hat{e}_2\}$ that allows us to express all vectors $\vec{v} \in \mathbb{R}^2$ as linear combinations of the basis vectors: $\vec{v} = v_1 \hat{e}_1 + v_2 \hat{e}_2$. The same vector \vec{v} corresponds to two different coordinate pairs, depending on which basis is used for the description: $\vec{v} = (v_x, v_y)$ in the basis $\{\hat{i}, \hat{j}\}$ and $\vec{v} = (v_1, v_2)$ in the basis $\{\hat{e}_1, \hat{e}_2\}$. We'll learn about **bases and their properties** the properties of bases in great detail in the coming chapters. The choice of basis plays a fundamental role in all aspects of linear algebra. ~~Bases relate the real world to its mathematical representation in terms of vector and matrix components.~~

In the text above, I explained that computing the product between a matrix and a vector $A\vec{x} = \vec{y}$ can be thought of as a linear transformation, with input \vec{x} and output \vec{y} . Any linear transformation (Section 5.1) can be represented (Section 5.2) as a multiplication by a matrix A . Conversely, every $m \times n$ matrix $A \in \mathbb{R}^{m \times n}$ can be thought of as performing a linear transformation $T_A: \mathbb{R}^n \rightarrow \mathbb{R}^m$. The equivalence between matrices and linear transformations allows us to identify certain matrix properties with properties of linear transformations. For example, the *column space* $\mathcal{C}(A)$ of the matrix A (the set of vectors that can be written as a combination of the columns of A) corresponds to the image space of the linear transformation T_A (the set of possible outputs of T_A).

The eigenvalues and eigenvectors of matrices (Section 6.1) allow us to describe the actions of matrices in a natural way. The set of eigenvectors of a matrix are special input vectors for which the action of the matrix is described as a *scaling*. When a matrix acts on one of its eigenvectors, the output is a vector in the same direction as the input vector scaled by a constant. The scaling constant is the *eigenvalue* (own value) associated with this eigenvector. By specifying all the eigenvectors and eigenvalues of a matrix, it is possible to obtain a complete description of what the matrix does. Thinking of

matrices in terms of their eigenvalues and eigenvectors is a powerful technique for describing their properties and has many applications.

Linear algebra is useful because linear algebra techniques can be applied to all kinds of “vector-like” objects. The abstract concept of a vector space (Section 6.3) captures precisely what it means for some class of mathematical objects to be “vector-like.” For example, the set of polynomials of degree at most two, denoted $P_2(x)$, consists of all functions of the form $f(x) = a_0 + a_1x + a_2x^2$. Polynomials are vector-like because it’s possible to describe each polynomial in terms of its coefficients (a_0, a_1, a_2) . Furthermore, the sum of two polynomials and the multiplication of a polynomial by a constant both correspond to vector-like calculations of coefficients. Once you realize polynomials are vector-like, you’ll be able to use **linear algebra concepts-notions** like *linear independence*, *dimension*, and *basis* when working with polynomials.

Useful linear algebra

One of the most useful skills you’ll learn in linear algebra is the ability to solve systems of linear equations. Many real-world problems are expressed as linear equations in multiple unknown quantities. You can solve for n unknowns simultaneously if you have a set of n linear equations that relate the unknowns. To solve this system of equations, eliminate the variables one by one using basic techniques such as substitution and subtraction (see Section ??); however, the procedure will be slow and tedious for many unknowns. If the system of equations is linear, it can be expressed as an *augmented matrix* built from the coefficients in the equations. You can then use the Gauss–Jordan elimination algorithm to solve for the n unknowns (Section 3.1). The key benefit of the augmented matrix approach is that it allows you to focus on the coefficients without worrying about the variable names. This saves time when you must solve for many unknowns. Another approach for solving n linear equations in n unknowns is to express the system of equations as a matrix equation (Section 3.2) and then solve the matrix equation by computing the matrix inverse (Section 3.5).

In Section 6.6 you’ll learn how to *decompose* a matrix into a product of simpler matrices. Matrix decompositions are often performed for computational reasons: certain problems are easier to solve on a computer when the matrix is expressed in terms of its simpler constituents. Other decompositions, like the decomposition of a matrix into its eigenvalues and eigenvectors, give you valuable information about the properties of the matrix. Google’s original PageRank algorithm for ranking webpages by “importance” can be explained as

2.12 Using the definition of the matrix vector product, we can imitate the action of each linear transformation T by choosing appropriate ~~coefficients~~ entries in the matrix. The top row corresponds to the first ~~coefficient~~ component of the output; the bottom row corresponds to the second ~~coefficient~~ component of the output. Observe that $BA = C$. The composite transformation of applying T_A followed by T_B (denoted $T_B \circ T_A$), is equivalent to the transformation T_C . Note $AB \neq BA$: the matrix product AB corresponds to composing the composition of the linear transformations in the opposite order $T_A \circ T_B$.

P2.13 Use the determinant properties to simplify these expressions:

$$\text{a) } \det(ABA^{-1}) \quad \text{b) } \frac{\det(AB)}{\det(B)} \quad \text{c) } \det(AB) - \det(BA)$$

2.13 a) $\det(B)$; b) $\det A$; c) 0.

P2.14 Suppose A , B , and C are unknown ~~3~~ 3×3 matrices such that $|A| = 2$, $|B| = 3$, and $|C| = 5$. Determine the values of the following expressions:

$$\text{a) } |AB| \quad \text{b) } |A^{-1}B| \quad \text{c) } |BC| \quad \text{d) } |ABC| \quad \text{e) } |2C|$$

Hint: Use the properties of the determinant operation.

2.14 a) 6; b) $\frac{3}{2}$; c) 15; d) 30; e) ~~40~~.

If we add α -times the first equation to the second equation, we obtain an equivalent system of equations:

$$\begin{aligned} ax + by &= c \\ (d + \alpha a)x + (e + \alpha b)y &= f + \alpha c. \end{aligned}$$

This is called a *row operation*: we added α -times the first row to the second row. Row operations change the coefficients of the system of equations, but leave the solution unchanged. Gauss–Jordan elimination is a systematic procedure for solving systems of linear equations using row operations.

Matrix product multiplication The product AB between matrices $A \in \mathbb{R}^{m \times \ell}$ and $B \in \mathbb{R}^{\ell \times n}$ is the matrix $C \in \mathbb{R}^{m \times n}$ whose **coefficients** entries c_{ij} are defined by the formula $c_{ij} = \sum_{k=1}^{\ell} a_{ik}b_{kj}$ for all $i \in [1, \dots, m]$ and $j \in [1, \dots, n]$. In Section 3.3, we’ll unpack this formula and learn **about** its intuitive interpretation: that computing $C = AB$ is computing all the dot products between the rows of A and the columns of B .

Determinant The determinant of a matrix A , denoted $\det(A)$, is an operation that gives us useful information about the linear independence of the rows of the matrix. The determinant is connected to many notions of linear algebra: linear independence, geometry of vectors, solving systems of equations, and matrix invertibility. We’ll discuss these aspects of determinants in Section 3.4.

Matrix inverse In Section 3.5, we’ll build upon our knowledge of Gauss–Jordan elimination, matrix products, and determinants to derive three different procedures for **computing** finding the matrix inverse A^{-1} .

3.1 Reduced row echelon form

In this section we’ll learn to solve systems of linear equations using the *Gauss–Jordan elimination* procedure. A system of equations can be represented as a matrix of coefficients. The Gauss–Jordan elimination procedure converts any matrix into its *reduced row echelon form* (RREF). We can use the RREF to easily find the solution (or solutions) of the system of equations.

Heads up: the material covered in this section requires your full-on, caffeinated attention, as the procedures you’ll learn are somewhat tedious. Gauss–Jordan elimination involves many repetitive

$$\boxed{A} \boxed{B} = \boxed{C}.$$

Again, we must *do the same* to both sides of the equation. To cancel A , we need to multiply by A^{-1} from the left:

$$\boxed{A^{-1}} \boxed{A} \boxed{B} = \boxed{A^{-1}} \boxed{C}.$$

When A^{-1} cancels with A , we obtain the final result

$$\boxed{B} = \boxed{A^{-1}} \boxed{C}.$$

This completes our lightning tour of matrix equations. There is really nothing new to learn here; just make sure you're aware that the *order* in which matrices are multiplied matters, and remember the general principle of "doing the same thing to both sides of the equation." Acting according to this principle is essential in all of math, and particularly important when manipulating **non-commutative noncommutative** operations as in matrix equations.

In the next section, we'll "zoom in" on matrix equations by examining the arithmetic operations performed on **coefficients during matrix-matrix entries during** multiplication.

Exercises

E3.4 Solve for X in the following matrix equations: **a)** $XA = B$, **b)** $ABCXD = E$, and **c)** $AC = XDC$. You can assume the matrices A , B , C , D , and E are all invertible.

3.4 a) $X = BA^{-1}$; **b)** $X = C^{-1}B^{-1}A^{-1}ED^{-1}$; **c)** $X = AD^{-1}$.

3.3 Matrix multiplication

Suppose we're given the matrices

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix},$$

Properties

Let A and B be two square matrices of the same dimension. The determinant operation has the following properties:

- $\det(AB) = \det(A) \det(B) = \det(B) \det(A) = \det(BA)$
- If $\det(A) \neq 0$, the matrix is invertible and $\det(A^{-1}) = \frac{1}{\det(A)}$
- $\det(A^T) = \det(A)$
- $\det(\alpha A) = \alpha^n \det(A)$, for an $n \times n$ matrix A
- $\det(A) = \prod_{i=1}^n \lambda_i$, where $\{\lambda_i\} = \text{eig}(A)$ are the eigenvalues of A

The effects of row operations on determinants

Recall the three row operations we used for the Gauss–Jordan elimination procedure:

- Add a multiple of one row to another row
- Swap two rows
- Multiply a row by a constant

We'll now describe the effects of these row operations on the value of the matrix determinant. In each case, we'll connect the effects of the row operation to the geometric interpretation of the determinant operation.

Add a multiple of one row to another row

Adding a multiple of one row of a matrix to another row does not change the determinant of the matrix.

$$\begin{vmatrix} r_1 \\ r_2 \\ r_3 \end{vmatrix} = \begin{vmatrix} r_1 + \alpha r_2 \\ r_2 \\ r_3 \end{vmatrix}$$

Figure 3.10: Row operations of the form $\mathcal{R}_\alpha : R_i \leftarrow R_i + \alpha R_j$ do not change the value of the matrix determinant.

This property follows from the fact that parallelepipeds with equal base enclosed between two parallel planes have the same volume even if they have different slants. This is ~~known as~~ called *Cavalieri's principle*.

It is easier to visualize Cavalieri's principle in two dimensions by considering two parallelograms with base b and different slants, enclosed between two parallel lines. The area of both parallelograms

Invertibility

Not all matrices are invertible. Keep this in mind, since teachers might try to trick you by asking you to find the inverse of a non-invertible matrix. Let's analyze how each procedure for computing the inverse fails when applied to a non-invertible matrix D . The inverse formula based on the determinant and the adjugate matrix is $D^{-1} = \frac{1}{\det(D)} \text{adj}(D)$. However, if the matrix D is not invertible, then $\det(D) = 0$ and the formula fails due to a divide-by-zero error. The row operations approach to computing the inverse will also fail. Starting from the extended array $[D | \mathbb{1}]$, you can apply all the row operations you want, but you'll never be able to obtain the identity matrix in the left half of the extended array. This is because the reduced row echelon form of a non-invertible matrix D has at least one row of zeros: $\text{rref}(D) \neq \mathbb{1}$. We'll discuss invertible matrices and their properties in Section 5.4. For now, be sure to remember the determinant invertibility test: if $\det(A) = 0$, then A is non-invertible, and if $\det(A) \neq 0$, then A is invertible.

Exercises

E3.9 For what values of α is the matrix $A = \begin{bmatrix} 2 & 3 \\ 4 & \alpha \end{bmatrix}$ invertible?

3.9 A^{-1} exists for all $\alpha \neq 6$.

E3.10 Compute A^{-1} where $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$.

3.10 $A^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$.

E3.11 Solve for x and y in the equation $\begin{bmatrix} 1 & 3 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

3.11 $x = -8$ and $y = 3$.

3.11 We can solve the equation $A \begin{bmatrix} x \\ y \end{bmatrix} = \vec{b}$ by multiplying the equation by the inverse A^{-1} . We find $A^{-1} = \begin{bmatrix} -2 & -3 \\ 1 & 1 \end{bmatrix}$ and $\begin{bmatrix} x \\ y \end{bmatrix} = A^{-1} \vec{b} = \begin{bmatrix} -8 \\ 3 \end{bmatrix}$.

E3.12 Show that for an $n \times n$ invertible matrix A , the determinant of the adjugate matrix is $|\text{adj}(A)| = (|A|)^{n-1}$.

Hint: Recall that $|A^{-1}| = \frac{1}{|A|}$ and $|\alpha A| = \alpha^n |A|$.

3.6 Computational problems

We've reached the problem section where you're supposed to practice all the computational techniques of linear algebra. This is not going to be the most exciting three hours of your life, but you'll get through it. You need to know how to solve computational problems by hand and apply the Gauss–Jordan elimination procedure; and you need to know how to multiply matrices, calculate determinants, and find matrix inverses. These computational techniques enable all the advanced procedures we'll develop later in the book. If you skip these practice problems, you'll have trouble later when it comes to mastering more advanced topics that rely on these basic matrix operations as building blocks. Do this important work now, and you'll be on your way to becoming fluent in linear algebra computations... plus, the rest of the book will be much more pleasant.

P3.1 Find the solution sets for the augmented matrices.

$$\text{a) } \left[\begin{array}{ccc|c} 1 & -1 & -2 & 1 \\ -2 & 2 & 4 & -2 \\ 3 & -3 & -6 & 3 \end{array} \right] \quad \text{b) } \left[\begin{array}{cccc|c} 2 & -2 & 3 & 2 & 2 \\ 0 & 0 & 5 & 3 & 3 \\ 6 & -6 & -1 & 0 & 0 \\ 6 & -6 & 9 & 6 & 6 \end{array} \right]$$

fixed

3.1 a) $\{(1, 0, 0) + s(1, 1, 0) + t(2, 0, 1), \forall s, t \in \mathbb{R}\}$; **b)** $\{(\frac{1}{10}, \frac{3}{5}, 0, 0) - \{(\frac{1}{10}, 0, \frac{3}{5}, 0) + \alpha(1, 1, 0, 0) + \beta(-\frac{1}{10}, 0, -\frac{3}{5}, 1), \forall \alpha, \beta \in \mathbb{R}\}$.

P3.2 Find the solutions to the systems of equations.

$$\text{a) } \left[\begin{array}{ccc|c} 2 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 4 & 2 & -2 & 0 \end{array} \right] \quad \text{b) } \left[\begin{array}{ccc|c} 2 & 0 & 1 & 5 \\ 1 & 4 & 2 & 2 \\ 0 & 2 & 1 & 1 \end{array} \right] \quad \text{c) } \left[\begin{array}{ccc|c} 1 & 1 & 2 & 2 \\ -2 & 2 & 4 & -8 \\ 3 & -3 & -6 & 3 \end{array} \right] \left[\begin{array}{ccc|c} 1 & 1 & 2 & 2 \\ 4 & -2 & -4 & -4 \\ 3 & -3 & -6 & -6 \end{array} \right]$$

3.2 a) $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + t \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \forall t \in \mathbb{R} \right\} \cup \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + t \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \forall t \in \mathbb{R} \right\}$. **b)** $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \\ 5 \end{bmatrix}$. **c)** $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + t \begin{bmatrix} 0 \\ -2 \\ 1 \end{bmatrix}, \forall t \in \mathbb{R} \right\} \cup \left\{ \begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix} + t \begin{bmatrix} 0 \\ -2 \\ 1 \end{bmatrix}, \forall t \in \mathbb{R} \right\}$.

P3.3 Consider an unknown matrix $A \in \mathbb{R}^{5 \times 5}$. You're told the system of linear equations $A\vec{x} = \vec{b}$ has an infinite number of solutions. What is the maximum rank of the matrix A ?

3.3 $\text{rank}(A) \leq 4$.

3.3 Since the system of equations $A\vec{x} = \vec{b}$ has an infinite number of solutions, the RREF of A must contain at least one row of zeros. Therefore, the rank of A can be at most $5 - 1 = 4$.

P3.4 Solve for C in the matrix equation $ABCD = AD$.

question
changed

3.4 $C = B^{-1}$.

3.4 First simplify the equation by multiplying with A^{-1} from the left, and with D^{-1} from the right, to obtain $BC = \mathbb{1}$. Now we can isolate C by multiplying with B^{-1} from the left. We obtain $B^{-1} = C$.

P3.5 Solve for the following matrix equations problems:

- a) Simplify the expression $MNB^{-1}BK^{-1}KN^{-1}M^{-2}L^{-1}S^{-1}SMK^2$.
- b) Simplify $J^{-3}K^2G^{-1}GK^{-3}J^2$.
- c) Solve for A in the equation $A^{-1}BNK = 2B^2B^{-1}NK$.
- d) Solve for Y in $SUNNY = SUN$.

You can assume all matrices are invertible.

fixed

3.5 a) $M^{-1}L^{-1}MKM^{-1}L^{-1}MK^2$; b) $J^{-3}K^{-1}J^2$; c) $A = \frac{1}{2}\mathbb{1}$; d) $Y = N^{-1}$.

P3.6 Solve for \vec{x} in $A\vec{x} = \vec{b}$, where $A = \begin{bmatrix} 1 & 0 & -3 \\ 2 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$ and $\vec{b} = (2, 2, 3)^T$.

Hint: Express the equation $A\vec{x} = \vec{b}$ as an augmented matrix.

3.6 $\vec{x} = (-7, -19, -3)^T$.

P3.7 Solve for \vec{x} in the equation $\vec{x} = \vec{d} + A\vec{x}$, where $A = \begin{bmatrix} 0 & 0.05 & 0.3 \\ 0.01 & 0 & 0.01 \\ 0.1 & 0 & 0 \end{bmatrix}$, and $\vec{d} = (25, 10, 14)^T$. Use `live.sympy.org` to perform the calculations.

Hint: Rewrite as $\mathbb{1}\vec{x} = \vec{d} + A\vec{x}$, then bring all the \vec{x} s to one side.

3.7 $\vec{x} = (30.64, 10.48, 17.06)$.

3.7 Start by rewriting the matrix equations as $(\mathbb{1} - A)\vec{x} = \vec{d}$, then solve for \vec{x} by hitting the equation with the appropriate inverse: $\vec{x} = (\mathbb{1} - A)^{-1}\vec{d}$. See `bit.ly/1hg44Ys` for the details of the calculation.

P3.8 Compute the product of three matrices:

$$\begin{bmatrix} 2 & 10 & -5 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 0 & 2 \\ 5 & 1 \\ -3 & -4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

3.8 $\begin{bmatrix} -2 & -2 \\ -15 & -15 \end{bmatrix}$.

P3.9 Consider the following three matrices:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \text{and} \quad H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Show that $HXH = Z$ and that $HZH = X$.

3.9 First rewrite H as $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ to simplify calculations. Then computing HXH gives us $HXH = \frac{1}{2} \left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} = Z$. The calculation for HZH is similar and it leads to X .

P3.10 Given an unknown variable $\alpha \in \mathbb{R}$ and the matrices

$$A = \begin{bmatrix} \cos(\alpha) & 1 \\ -1 & -\sin(\alpha) \end{bmatrix}; \quad B = \begin{bmatrix} \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) \end{bmatrix}; \quad C = \begin{bmatrix} 1 & -\cos(\alpha) \\ \sin(\alpha) & 1 \end{bmatrix},$$

compute the value of **a)** $A^2 + B^2$, **b)** $A^2 + C$, and **c)** $A^2 + C - B^2$. Give your answer in terms of α and use the double-angle formulas as needed.

P3.11 Find the determinants of the following matrices.

$$\text{a) } \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix} \qquad \text{b) } \begin{bmatrix} 0 & 5 & 3 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \text{c) } \begin{bmatrix} 1 & 2 & 0 \\ 3 & 1 & 1 \\ 4 & -2 & 0 \end{bmatrix}$$

3.11 **a)** -3 ; **b)** 0 ; **c)** 10 .

P3.12 Find the determinants of the matrices.

$$A = \begin{bmatrix} 3 & -1 & 5 & 2 \\ 0 & 2 & 2 & -3 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix} \qquad B = \begin{bmatrix} 2 & -1 & 0 & -3 & 2 \\ 0 & 1 & 1 & 3 & 0 \\ 1 & 4 & 0 & 0 & -1 \\ 3 & -2 & 3 & 1 & 0 \\ -1 & 0 & -1 & 0 & 2 \end{bmatrix}$$

3.12 $\det(A) = -48$; $\det(B) = 13$.

P3.13 Determine if the following sets of vectors are linearly dependent or linearly independent.

$$\text{a) } \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\} \quad \text{b) } \{(1, 1, 0), (1, 0, 1), (1, 3, -2)\} \\ \text{c) } \{(1, 2, 3, 4), (-1, -2, 1, 0), (0, 0, 1, 1), (1, 0, 0, 1)\}$$

3.13 **a)** Independent; **b)** Dependent; **c)** Dependent.

3.13 We can determine if the sets of vectors are linearly independent by combining them to form a matrix then computing this matrix's determinant. If the determinant of the matrix is nonzero, the vectors are linearly independent.

P3.14 Find the area of a parallelogram that has vectors $\vec{v} = (3, -5)$ and $\vec{w} = (1, -1)$ as its sides.

Hint: Use the formula from Section 3.4 (page 73).

3.14 Area = 2.

P3.15 Find the volume of the parallelepiped that has the vectors $\vec{u} = (2, 0, 1)$, $\vec{v} = (1, -1, 1)$, and $\vec{w} = (0, 2, 3)$ as sides. [See Figure 3.9 for an illustration.](#)

3.15 Volume = 8.

P3.16 Suppose M and N are unknown 4×4 matrices with $|M| = -2$ and $|N| = 7$. Compute the values of these determinant expressions:

3.21 a) The inverse doesn't exist; b) $\begin{bmatrix} 5 & -2 \\ -2 & 1 \end{bmatrix}$; c) $\begin{bmatrix} 2 & -\frac{3}{2} \\ -1 & 1 \end{bmatrix}$.

P3.22 Given the matrix equation $AB = C$, where A and C are 2×2 matrices, find the matrix B .

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 7 \end{bmatrix} \qquad C = \begin{bmatrix} 3 & 2 \\ 1 & -4 \end{bmatrix}$$

3.22 $B = \begin{bmatrix} -17 & -30 \\ 5 & 8 \end{bmatrix}$.

P3.23 Find the inverses of the matrices $A = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 1 & 1 \\ 0 & -2 & -1 \end{bmatrix}$ and $B = \begin{bmatrix} 0 & -3 & 2 & 4 \\ 1 & -1 & 1 & -1 \\ 2 & 4 & 0 & -2 \\ 3 & 0 & 1 & 0 \end{bmatrix}$.

3.23 $A^{-1} = \begin{bmatrix} -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{2}{3} & \frac{1}{3} & -\frac{5}{3} \\ \frac{4}{3} & -\frac{2}{3} & \frac{7}{3} \end{bmatrix}$ and $B^{-1} = \frac{1}{21} \begin{bmatrix} -3 & -5 & \frac{-21}{6} & 11 \\ 3 & -2 & 7 & -4 \\ 9 & 15 & \frac{21}{2} & -12 \\ 3 & -9 & 0 & 3 \end{bmatrix}$.

P3.24 Prove that the zero matrix A has no inverse.

3.24 Zero matrix has the $\det(A) = 0$. We have $A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$. We cannot divide by zero, so the zero matrix has no inverse.

P3.25 Obtain the matrices of cofactors for the following matrices.

$$A = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 1 & 1 \\ 0 & -2 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 0 & 1 \\ 3 & -1 & -3 \\ 0 & -4 & -2 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 0 & -2 & 0 \\ -1 & 2 & 1 & -2 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

3.25 $C_A = \begin{bmatrix} 1 & 2 & -4 \\ -2 & -1 & 2 \\ 1 & 5 & -7 \end{bmatrix}$; $C_B = \begin{bmatrix} -10 & 6 & -12 \\ -4 & -10 & 20 \\ 1 & 18 & -5 \end{bmatrix}$; $C_C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & -2 & 2 & -2 \\ -4 & 4 & -4 & 4 \\ -8 & 8 & -8 & 8 \end{bmatrix}$.

P3.26 Implement the formula $A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$ for the case of 3×3 matrices using a spreadsheet application like [LibreOffice](#), OpenOffice, Excel, or Google [DoesSheets](#). Assume the [coefficients](#) [entries](#) of the matrix are specified in the top right corner of the spreadsheet: A1:C3. Start by writing the formula for computing the determinant and the matrix of cofactors, then combine these partial calculations to obtain the nine [coefficients](#) [entries](#) of the matrix inverse. Test that your formula is correct by finding the inverse of $A = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 0 & 1 \\ 0 & -2 & -1 \end{bmatrix}$. Compare your formula's output with the built-in function `=MINVERSE(A1:C3)`.

3.26 $A^{-1} = \begin{bmatrix} -1 & 1 & -2 \\ -1 & \frac{1}{2} & -\frac{5}{2} \\ 2 & -1 & 4 \end{bmatrix}$.

3.26 See bit.ly/matinvxls for the solution.

P3.27 Find the values a , b , c , and d that satisfy the equation

$$\begin{bmatrix} 1 & 3 \\ -2 & -1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 3 & -5 \\ 4 & 0 \end{bmatrix}.$$

3.27 $a = -3, b = 1, c = 2, d = -2$.

3.27 Find the inverse of $\begin{bmatrix} 1 & 3 \\ -2 & -1 \end{bmatrix}$, then multiply both sides of the equation by the inverse to isolate the matrix of unknowns $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

P3.28 Given the constraints $a = g, e = b = f$, and $c = d = h$, find a choice of the variables a, b, c, d, e, f, g, h that satisfies the matrix equation:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} -2 & -3 \\ 0 & 2 \end{bmatrix}.$$

3.28 $\begin{bmatrix} 1 & -1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} -1 & -1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -3 \\ 0 & 2 \end{bmatrix}$.

3.28 First we calculate $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & df+dh \end{bmatrix} = \begin{bmatrix} 2ae & ae+ec \\ ce+ca & c^2+ec \end{bmatrix}$. We proceed by comparing the two matrices component-by-componententry by entry. Observe $2ae = -2$, which implies $ae = -1$, which allows us to simplify $ae + ec = -3$ to $ec = -2$. The equation $ec + ca = 0$ implies $ca = 2$. Finally, using $ce + c^2 = 2$ and $ec = -2$, we find $c = 2$. Therefore $c = d = h = 2$, $a = g = 1$, and $e = b = f = -1$.

P3.29 Given the matrix $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, explain how you can obtain the matrices $B = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} \\ \beta a_{21} & \beta a_{22} \end{bmatrix}$ and $C = \begin{bmatrix} \alpha a_{11} & \beta a_{12} \\ \alpha a_{21} & \beta a_{22} \end{bmatrix}$.

Hint: Use matrix multiplication (from the left and from the right).

3.29 $B = SA$ and $C = AS$, where $S = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}$.

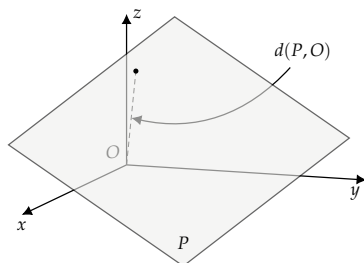


Figure 4.9: The closest distance between the plane P and the origin.

new math operations are required. Instead, we'll learn how to use a combination of vector subtraction, vector length, and the dot product to compute distances. Each distance function $d(\cdot, \cdot)$ corresponds to an abstract procedure with one or two steps which can be described using a vector diagram. Projections play a key role in projective geometry, so we'll learn about them in detail in the next section.

Exercises

E4.1 Find the distance between the plane P with geometric equation $(1, 1, 1) \cdot [(x, y, z) - (4, 5, 6)] = 0$ and the origin.

4.1 $d(P, O) = 5\sqrt{3} \approx 8.66$.

E4.2 Find the general equation of the line that passes through the points $(0, 5)$ and $(6, -7)$ in \mathbb{R}^2 .

4.2 $2x + y = 5$.

E4.3 Given the point $r = (1, 3, 0)$, the line $\ell: \{(0, 0, 2) + t(1, 1, 0), t \in \mathbb{R}\}$ $\ell: \{(0, 0, 2) + t(1, -1, 0), t \in \mathbb{R}\}$, the plane $P: x + y + z = 1$, and the origin $O = (0, 0, 0)$, compute the following closest distances:

a) $d(r, O)$ b) $d(\ell, O)$ c) $d(P, O)$ d) $d(r, \ell)$ e) $d(r, P)$ f) $d(\ell, P)$

question
on
change
d

Hint: Draw a diagram and find the closest distances visually. Label the point $p_\ell = (0, 0, 2)$, which is an arbitrary point in the line, and the point $p_P = (1, 0, 0)$, which is an arbitrary point in the plane. Don't look for a one-size-fits-all formula for the different cases; derive the appropriate formula for each case starting from the basic projection operations Π_ℓ and Π_P .

4.3 a) $\sqrt{10}$; b) 2; c) $\frac{\sqrt{3}}{3}$; d) $\frac{\sqrt{62}\sqrt{3}}{3}$; e) $\sqrt{3}$; f) $\frac{\sqrt{3}}{3}$.

4.3 a) This is the norm of $\|r\|$. b) To find a vector from the origin to the closest point on the line ℓ , find a vector to any point on the line and

subtract the part of the vector that is parallel to the line. The length of this perpendicular-only vector is the closest distance $d(\ell, O)$. **c)** Start with an arbitrary point in the plane, say $p_P = (1, 0, 0)$. Then compute the length of the projection of p_P in the normal direction to obtain a vector to the point closest to the origin in the plane P . The length of this vector is equal to $d(P, O)$. **d)** The procedure is analogous to part (b), but we'll use a vector $\vec{u} = (0, 0, 2) - (1, 3, 0)$, which starts at r and ends on a point on the line ℓ . The part of the vector \vec{u} that is perpendicular to the line ℓ is computed as $\Pi_{\ell^\perp}(\vec{u}) = \vec{u} - \Pi_\ell(\vec{u})$. The closest distance between r and ℓ is the length of this vector, $\|\Pi_{\ell^\perp}(\vec{u})\|$. **e)** The procedure is analogous to part (c), but we start with the vector $\vec{v} = (1, 0, 0) - (1, 3, 0)$, which starts at r and ends on a point in the plane P . The closest distance between r and P is $\|\Pi_{\tilde{n}_P}(\vec{v})\|$. **f)** The line ℓ is parallel to the plane P . Define a vector that starts at some point on the line and ends at some point in the plane $\vec{w} = (1, 0, 0) - (0, 0, 2)$; then compute $\|\Pi_{\tilde{n}_P}(\vec{w})\|$. ~~See for calculations.~~

4.2 Projections

In this section we'll learn to compute projections of vectors onto lines and planes. Given an arbitrary vector, we'll find how much of this vector points in a given direction (projection onto a line). We'll also find the part of the vector that lies in some plane (projection onto a plane). The dot product, $\vec{u} \cdot \vec{v} \equiv u_1v_1 + u_2v_2 + u_3v_3$, $\vec{u} \cdot \vec{v} = u_1v_1 + u_2v_2 + u_3v_3$, will play a central role in these calculations.

Each projection formula corresponds to a vector diagram. Vector diagrams, also known as “picture proofs,” are used to describe the precise sequence of operations for computing a projection. Focussing on vector diagrams makes it much easier to understand projection and distance formulas. Indeed, the pictures in this section are a heck of a lot more important than the formulas. Be sure you understand each vector diagram, and don't worry about memorizing the corresponding formula. You can easily reproduce the formula by starting from the vector diagram.

Concepts

- $S \subseteq \mathbb{R}^n$: S is a *vector subspace* of \mathbb{R}^n . ~~In this chapter~~In this section, we assume $S \subseteq \mathbb{R}^3$. The subspaces of \mathbb{R}^3 are lines ℓ and planes P that pass through the origin.
- S^\perp : the orthogonal space complement to S , $S^\perp \equiv \{\vec{w} \in \mathbb{R}^n \mid \vec{w} \cdot S = 0\}$. $S^\perp \stackrel{\text{def}}{=} \{\vec{w} \in \mathbb{R}^n \mid \vec{w} \cdot S = 0\}$. The symbol $^\perp$ stands for *perpendicular*

to.

- Π_S : the *projection* onto the subspace S .
- Π_{S^\perp} : the projection onto the **orthogonal space subspace** S^\perp .

Definitions

Let S be a *vector subspace* of \mathbb{R}^n , denoted $S \subseteq \mathbb{R}^n$. In this section, we'll focus on the subspaces of the vector space \mathbb{R}^3 because they are easy to visualize and understand intuitively. The vector subspaces of \mathbb{R}^3 are lines and planes that pass through the origin. We defer the general discussion of subspaces in n dimensions until Section 4.4.

The projection operation onto the subspace S is a linear transformation that takes as inputs vectors in \mathbb{R}^3 , and produces outputs in the subspace S :

$$\Pi_S : \mathbb{R}^3 \rightarrow S.$$

The transformation Π_S , pronounced “projection onto S ,” cuts off all parts of the input that do not lie within the subspace S . We can understand Π_S by analyzing its action for different inputs:

- If $\vec{v} \in S$, then $\Pi_S(\vec{v}) = \vec{v}$.
- If $\vec{w} \in S^\perp$, then $\Pi_S(\vec{w}) = \vec{0}$.
- Linearity and the above two conditions imply that, for any vector $\vec{u} = \alpha\vec{v} + \beta\vec{w}$ with $\vec{v} \in S$ and $\vec{w} \in S^\perp$, we have

$$\Pi_S(\vec{u}) = \Pi_S(\alpha\vec{v} + \beta\vec{w}) = \alpha\vec{v}.$$

The *orthogonal subspace* to S is the set of vectors that are perpendicular to all vectors in S :

$$S^\perp \equiv \stackrel{\text{def}}{=} \{ \vec{w} \in \mathbb{R}^3 \mid \vec{w} \cdot \vec{s} = 0, \forall \vec{s} \in S \}.$$

The transformation Π_S “projects” to the space S in the sense that, no matter which vector \vec{u} you start from, applying the projection Π_S results in a vector that is in S :

$$\forall \vec{u} \in \mathbb{R}^3, \quad \Pi_S(\vec{u}) \in S.$$

All parts of \vec{u} in the *perp*-space S^\perp were killed by Π_S . Meet Π_S —the *S-perp* killer.

We can split the set of all vectors \mathbb{R}^3 into two disjoint sets: vectors entirely contained in S and vectors perpendicular to S . We say \mathbb{R}^3 decomposes into the *direct sum* of the subspaces S and S^\perp :

$$\mathbb{R}^3 = S \oplus S^\perp.$$

Any vector $\vec{u} \in \mathbb{R}^3$ can be split into an S -part $\vec{v} = \Pi_S(\vec{u})$ and a S^\perp -part $\vec{w} = \Pi_{S^\perp}(\vec{u})$, such that

$$\vec{u} = \vec{v} + \vec{w}.$$

A defining property of projections is that they are *idempotent operations*, meaning it doesn't matter if you project a vector once, twice, or a million times; the result will always be the same:

$$\Pi_S(\vec{u}) = \Pi_S(\Pi_S(\vec{u})) = \Pi_S(\Pi_S(\Pi_S(\vec{u}))) = \dots$$

Once you project a vector onto the subspace S , any further projections to S have no effect.

In the remainder of this section, we'll derive formulas for projections onto lines and planes that pass through the origin.

Projection onto a line

Consider the line ℓ passing through the origin with direction vector \vec{v} :

$$\ell : \{(x, y, z) \in \mathbb{R}^3 \mid (x, y, z) = \vec{0} + t\vec{v}, t \in \mathbb{R}\}.$$

The projection onto ℓ for an arbitrary vector $\vec{u} \in \mathbb{R}^3$ is given by the formula

$$\Pi_\ell(\vec{u}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{v}\|^2} \vec{v}.$$

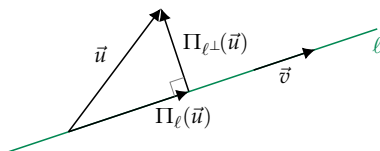


Figure 4.10: The vector \vec{u} can be decomposed into the sum of two projections defined with respect to the line ℓ . The projection $\Pi_\ell(\vec{u})$ is parallel to the line ℓ , while the projection $\Pi_{\ell^\perp}(\vec{u})$ is perpendicular to the line ℓ .

The orthogonal **space-complement** to the line ℓ consists of all vectors perpendicular to the direction vector \vec{v} . Mathematically speaking,

$$\ell^\perp : \{(x, y, z) \in \mathbb{R}^3 \mid (x, y, z) \cdot \vec{v} = 0\}.$$

Recognize that the equation $(x, y, z) \cdot \vec{v} = 0$ defines a *plane*. The orthogonal **space-complement** to a line ℓ with direction vector \vec{v} is a plane with normal vector \vec{v} . Makes sense, yes?

4.4 Vector spaces

We're about to shift our attention from individual vectors to entire sets of vectors. We're entering the territory of *vector spaces*. For instance, the set of all possible three-dimensional vectors is denoted \mathbb{R}^3 , and is a type of *vector space*. A vector space consists of a set of vectors and all linear combinations of these vectors. This means if the vectors \vec{v}_1 and \vec{v}_2 are part of some vector space, then so is the vector $\alpha\vec{v}_1 + \beta\vec{v}_2$ for any α and β . A *vector subspace* consists of a subset of all possible vectors. The vector subspaces of \mathbb{R}^3 are lines and planes that pass through the origin.

Since vector spaces and subspaces play a central role in many areas of linear algebra, you'll want to learn about the properties of vector spaces and develop your vocabulary for describing them.

By using the language of vector spaces, you'll be able to describe certain key properties of matrices. The [four fundamental subspaces](#) associated with a matrix A are its *column space* $\mathcal{C}(A)$, its *row space* $\mathcal{R}(A)$, its *null space* $\mathcal{N}(A)$, and its *left null space* $\mathcal{N}(A^T)$. Let's now define these vector spaces and discuss how they help us understand the solutions to the matrix equation $A\vec{x} = \vec{b}$, and the properties of the linear transformation $T_A(\vec{x}) = A\vec{x}$.

Definitions

- V : a *vector space*
- \vec{v} : a *vector*. We use the notation $\vec{v} \in V$ to indicate the vector \vec{v} is part of the vector space V .
- W : a *vector subspace*. We use the notation $W \subseteq V$ to indicate the vector space W is a subspace of the vector space V .
- *span*: the span of a set of vectors is the set of vectors that can be constructed as linear combinations of these vectors:

$$\text{span}(\vec{v}_1, \dots, \vec{v}_n) \stackrel{\text{def}}{=} \{ \vec{v} \in V \mid \vec{v} = \alpha_1\vec{v}_1 + \dots + \alpha_n\vec{v}_n, \alpha_i \in \mathbb{R} \}.$$

For every matrix $M \in \mathbb{R}^{m \times n}$, we define the following *fundamental vector spaces* associated with the matrix M :

- $\mathcal{R}(M) \subseteq \mathbb{R}^n$: the *row space* of the matrix M consists of all possible linear combinations of the rows of the matrix M .
- $\mathcal{C}(M) \subseteq \mathbb{R}^m$: the *column space* of the matrix M consists of all possible linear combinations of the columns of the matrix M .
- $\mathcal{N}(M) \subseteq \mathbb{R}^n$: the *null space* of M is the set of vectors that go to the zero vector when multiplying M from the right:
 $\mathcal{N}(M) = \{ \vec{v} \in \mathbb{R}^n \mid M\vec{v} = \vec{0} \}$

A real-life situation

You walk into class one day and are caught completely off guard by a surprise quiz—wait, let’s make it a mini-exam for emotional effect. Although you’ve read a chapter or two in the book, you’ve been “busy” and are totally unprepared for this exam. The first question asks you to “find the solution of the *homogeneous* system of equations and the *non-homogeneous* system of equations.” You rack your brain, but the only association with homogeneity that comes to mind is the homogenized milk you had for breakfast. Oh, there’s more: the question also asks you to “state whether each of the solutions obtained is a *vector space*.” As you stare at the page, the words and equations begin to blur and panic sets in.

Don’t fear! Look at the problem again. You don’t know what the heck a homogeneous system of equations is, but you sure as heck know how to solve systems of equations. You solve the given system of equations $A\vec{x} = \vec{b}$ by building the augmented matrix $[A \mid \vec{b}]$ and computing its reduced row echelon form using row operations. You obtain the solution set $\vec{x} = \{\vec{v} \in V \mid \vec{v} = \vec{c} + t\vec{v}_n, \forall t \in \mathbb{R}\}$, where \vec{c} is the particular solution and \vec{v}_n is a vector that spans the null space of A .

Next, you ponder the “vector space” part of the question. You notice the solution set to the *system* of equations $A\vec{x} = \vec{b}$ isn’t a vector space since it doesn’t pass through the origin. However, the solution set to the equation $A\vec{x} = \vec{0}$ is a vector space $\{\vec{v} \in V \mid \vec{v} = t\vec{v}_n, \forall t \in \mathbb{R}\} = \text{span}(\vec{v}_n)$. Suddenly it clicks: a *homogeneous* system of equations must be the system of equations $A\vec{x} = \vec{0}$, in which the constants on the right side of the equation are all zero. The term *homogeneous* kind of makes sense; all the constants of the right side have the same value $b_1 = b_2 = \dots = 0$. The solution to the non-homogeneous system of equations $A\vec{x} = \vec{b}$ is the set $\{\vec{c} + s\vec{v}_n, \forall s \in \mathbb{R}\}$, which is not a vector space. The solution to the homogeneous system of equations $A\vec{x} = \vec{0}$ is $\{\vec{v} \in V \mid \vec{v} = t\vec{v}_n, \forall t \in \mathbb{R}\}$, which is a vector space. Well done!

Matrix fundamental ~~spaces~~subspaces

We now define four *fundamental* ~~vector spaces~~subspaces associated with a matrix $M \in \mathbb{R}^{m \times n}$.

- The column space $\mathcal{C}(M)$ is the span of the columns of the matrix. The column space consists of all possible output vectors the matrix can produce when multiplied by a vector from the

right:

$$\mathcal{C}(M) \stackrel{\text{def}}{=} \{ \vec{w} \in \mathbb{R}^m \mid \vec{w} = M\vec{v} \text{ for some } \vec{v} \in \mathbb{R}^n \}.$$

- The null space $\mathcal{N}(M)$ of a matrix $M \in \mathbb{R}^{m \times n}$ consists of all vectors the matrix M sends to the zero vector:

$$\mathcal{N}(M) \stackrel{\text{def}}{=} \{ \vec{v} \in \mathbb{R}^n \mid M\vec{v} = \vec{0} \}.$$

The null space is sometimes called the *kernel* of the matrix.

- The row space $\mathcal{R}(M)$ is the span of the rows of the matrix. We obtain linear combinations of the rows by multiplying the matrix with an m -dimensional vector from the left:

$$\mathcal{R}(M) \stackrel{\text{def}}{=} \{ \vec{v} \in \mathbb{R}^n \mid \vec{v} = \vec{w}^T M \text{ for some } \vec{w} \in \mathbb{R}^m \}.$$

Note, we used the transpose T to transform \vec{w} to a row vector.

- The left null space $\mathcal{N}(M^T)$ of a matrix $M \in \mathbb{R}^{m \times n}$ consists of all vectors the matrix M sends to the zero vector when multiplied from the left:

$$\mathcal{N}(M^T) \stackrel{\text{def}}{=} \{ \vec{w} \in \mathbb{R}^m \mid \vec{w}^T M = \vec{0}^T \}.$$

These vector spaces are called *fundamental* because they describe important properties of the matrix M . Recall that matrix equations can be used to represent systems of linear equations, as well as linear transformations. A solid understanding of **fundamental spaces of a matrix** the fundamental subspaces leads to a solid understanding of linear equations and linear transformations.

Matrices and systems of linear equations

The null space $\mathcal{N}(M)$ corresponds to the solution set of the matrix equation $M\vec{x} = \vec{0}$. If a matrix has a nonempty null space, the system of equations corresponding to $M\vec{x} = \vec{b}$ has an infinite solution set. Indeed, we can write the solution of $M\vec{x} = \vec{b}$ as a *particular solution* \vec{c} plus all possible vectors in the null space of M :

$$\vec{x} = \vec{c} + \text{span}(\vec{v}_1, \dots, \vec{v}_k), \quad \text{where} \quad \text{span}(\vec{v}_1, \dots, \vec{v}_k) \stackrel{\text{def}}{=} \mathcal{N}(M).$$

We can verify this claim as follows. Suppose $\vec{x} = \vec{c}$ is a solution to the equation $M\vec{x} = \vec{b}$. Consider the vector $\vec{x} = \vec{c} + \alpha_1 \vec{v}_1 + \dots + \alpha_k \vec{v}_k$ $\vec{x} = \vec{c} + \alpha_1 \vec{v}_1 + \dots$ which contains \vec{c} and some arbitrary linear combination of vectors

linearly independent, the same vector \vec{v} can be represented by many different linear combinations of the form

$$\vec{v} = v'_1 \vec{u}_1 + v'_2 \vec{u}_2 + \cdots + v'_m \vec{u}_m.$$

We cannot identify \vec{v} with a *unique* set of coefficients v'_1, v'_2, \dots, v'_m , therefore vectors are not represented faithfully by ~~their coefficients~~ the coefficients in the linear combination.

Another reason we prefer to describe V in terms of a basis is because we can immediately see the vector space V is n -dimensional, since there are n vectors in the basis for V .

Definitions

- $B = \{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n\}$. A *basis* for an n -dimensional vector space S is a set of n linearly independent vectors that span S . Any vector $\vec{v} \in S$ can be written as a linear combination of the basis vectors:

$$\vec{v} = v_1 \vec{e}_1 + v_2 \vec{e}_2 + \cdots + v_n \vec{e}_n.$$

A basis for an n -dimensional vector space contains exactly n vectors.

- $\dim(S)$: the dimension of the vector space S is equal to the number of vectors in a basis for S .

Recall the four *fundamental* ~~spaces~~subspaces of a matrix $M \in \mathbb{R}^{m \times n}$ we defined in the previous section:

- $\mathcal{R}(M) \subseteq \mathbb{R}^n$: the *row space* of the matrix M that consists of all possible linear combinations of the rows of the matrix M .
- $\mathcal{N}(M) \subseteq \mathbb{R}^n$: the *null space* of the matrix contains all the vectors that become the zero vector when multiplied by M :

$$\mathcal{N}(M) \stackrel{\text{def}}{=} \{ \vec{v} \in \mathbb{R}^n \mid M\vec{v} = \vec{0} \}.$$

- $\mathcal{C}(M) \subseteq \mathbb{R}^m$: the *column space* of the matrix M that consists of all possible linear combinations of the columns of the matrix M .
- $\mathcal{N}(M^T) \subseteq \mathbb{R}^m$: the *left null space* of the matrix contains all the vectors that become the zero vector when multiplying M from the left:

$$\mathcal{N}(M^T) \stackrel{\text{def}}{=} \{ \vec{w} \in \mathbb{R}^m \mid \vec{w}^T M = \vec{0}^T \}.$$

Bases for the fundamental ~~spaces~~ subspaces of matrices

Performing the Gauss–Jordan elimination procedure on a matrix A has the effect of distilling a basis for its row space $\mathcal{R}(A)$. How do we find bases for the other fundamental ~~spaces~~ subspaces of a matrix? In this section, we’ll learn about a useful shortcut for computing bases for the column space $\mathcal{C}(A)$ and the null space $\mathcal{N}(A)$ of a matrix, starting from the reduced row echelon form of the matrix. Sorry, there is no shortcut for finding the left null space—we’ll have to use the transpose operation to obtain A^T and then find its null space $\mathcal{N}(A^T)$.

Pay careful attention to the locations of the pivots (leading ones) in the RREF of A , because they play an important role in the procedures described below.

Basis for the row space

The row space $\mathcal{R}(A)$ of a matrix A is defined as the space of all vectors that can be written as linear combinations of the rows of A . To find a basis for $\mathcal{R}(A)$, we use the Gauss–Jordan elimination procedure:

1. Perform row operations to find the RREF of A .
2. The nonzero rows in the RREF of A form a basis for $\mathcal{R}(A)$.

Basis for the column space

To find a basis for the column space $\mathcal{C}(A)$ of a matrix A , we need to determine which columns of A are linearly independent. To find the linearly independent columns of A , follow these steps:

1. Perform row operations to find the RREF of A .
2. Identify the columns that contain pivots (leading ones).
3. The corresponding columns in the original matrix A form a basis for the column space of A .

This procedure works because elementary row operations do not change the independence relations between the columns of the matrix. If two columns are linearly independent in the RREF of A , then these columns are also linearly independent in the original matrix A .

Note that the column space of the matrix A corresponds to the row space of the matrix transposed A^T . From this fact, we derive another procedure for finding a basis for the column space of a matrix A : we can use the find-a-basis-for-the-row-space procedure on A^T .

4. We now clear the entry below the pivot using $R_3 \leftarrow R_3 + 4R_2$:

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

5. The final simplification step is to clear the -1 in the first row using $R_1 \leftarrow R_1 + R_2$:

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

Now that we have the RREF of the matrix, we can answer the questions like professionals.

Before we find bases for the fundamental spaces subspaces of A , let's first do some basic dimension counting. Observe that the matrix has just two pivots. We say $\text{rank}(A) = 2$. This means both the row space and the column spaces are two-dimensional.

Recall the equation $n = \text{rank}(A) + \text{nullity}(A)$, which we saw in the previous section. The right space \mathbb{R}^3 splits into two types of vectors: those in the row space of A and those in the null space. Since we know the row space is two-dimensional, we can deduce the dimension of the null space: $\text{nullity}(A) = \text{dim}(\mathcal{N}(A)) = n - \text{rank}(A) = 3 - 2 = 1$ nullity

Now let's answer the questions posed in the problem. The row space of A consists of the two nonzero vectors in the RREF of A :

$$\mathcal{R}(A) = \text{span}((1, 0, -1), (0, 1, -1)).$$

To find the column space of A , observe that the first and second columns contain the pivots in the RREF of A . If they do, then the first two columns of the original matrix A form a basis for the column space of A :

$$\mathcal{C}(A) = \text{span}\left(\begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -4 \\ 1 \\ -6 \end{bmatrix}\right).$$

Let's now find an expression for the null space of A . First, observe that the third column does not contain a pivot. No pivot indicates that the third column corresponds to a free variable; it can take on any value, which we write $x_3 = t$, $t \in \mathbb{R}$. We want to give a description of all vectors $(x_1, x_2, t)^T$ that satisfy the system of equations:

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \begin{array}{rcl} 1x_1 - 1t & = & 0 \\ 1x_2 - 1t & = & 0 \\ 0 & = & 0. \end{array}$$

We find the values of x_1 , x_2 , and x_4 in terms of t and obtain

$$\mathcal{N}(B) = \left\{ \begin{bmatrix} 2t \\ -t \\ t \\ 0 \end{bmatrix}, t \in \mathbb{R} \right\} = \text{span} \left(\begin{bmatrix} 2 \\ -1 \\ 1 \\ 0 \end{bmatrix} \right).$$

Discussion

Dimensions

For an $m \times n$ matrix $M \in \mathbb{R}^{m \times n}$ the row space and the ~~column~~-null space consist of vectors with n components, while the column space and the left null space consist of vectors with m components.

Don't confuse the number of components of vectors in a vector space with the *dimension* of the space. Suppose we're given a matrix $M \in \mathbb{R}^{5 \times 10}$ with five rows and 10 columns, and the RREF of M contains three pivots. We say the *rank* of the matrix is 3, which means the row space of M is three-dimensional. A basis for the row space of M contains three vectors, each vector having 10 components. The null space of the matrix is seven-dimensional ($10 - 3 = 7$) and consist of vectors with 10 components. The column space of the matrix is also three-dimensional ($\dim(\mathcal{R}(M)) = \dim(\mathcal{C}(M))$). A basis for the column space of M consists of three vectors with five components. The left null space of M is two-dimensional ($5 - 3 = 2$) and is spanned by vectors with five components.

Importance of bases

The procedures for identifying bases are somewhat technical and potentially boring, but they are of great practical importance. To illustrate the importance of a basis, consider a scenario in which you're given a description of the xy -plane P_{xy} as the span of *three* vectors:

$$P_{xy} = \text{span}((1, 0, 0), (0, 1, 0), (1, 1, 0)).$$

The above definition of P_{xy} says that any point $p \in P_{xy}$ can be written as a linear combination of the form

$$p = a(1, 0, 0) + b(0, 1, 0) + c(1, 1, 0),$$

for some coefficients a , b , and c . This representation of P_{xy} is misleading. It might make us think (erroneously) that P_{xy} is three-dimensional, since it takes three coefficients to describe points in P_{xy} .

Chapter 5

Linear transformations

Linear transformations are a central idea of linear algebra—they form the cornerstone that connects all the seemingly unrelated concepts we’ve studied so far. We previously introduced linear transformations, informally describing them as “vector functions.” In this chapter, we’ll formally define linear transformations, describe their properties, and discuss their applications.

In Section 5.2, we’ll learn how matrices can be used to *represent* linear transformations. We’ll show the matrix representations of important types of linear transformations like projections, reflections, and rotations. Section 5.3 discusses the relation between bases and matrix representations. We’ll learn how the bases chosen for the input and output spaces determine the ~~coefficients~~ entries of matrix representations. A single linear transformation can correspond to many different matrix representations, depending on the choice of bases for the input and output spaces.

Section 5.4 discusses and characterizes the class of *invertible linear transformations*. This section serves to connect several topics we covered previously: linear transformations, matrix representations, and the fundamental ~~vector spaces~~ subspaces of matrices.

5.1 Linear transformations

Linear transformations take vectors as inputs and produce vectors as outputs. A transformation T that takes n -dimensional vectors as inputs and produces m -dimensional vectors as outputs is denoted $T: \mathbb{R}^n \rightarrow \mathbb{R}^m$.

The class of linear transformations includes most of the useful transformations of analytical geometry: stretchings, projections, reflections, rotations, and combinations of these. Since linear trans-

Linear transformations as black boxes

Suppose someone gives you a black box that implements the linear transformation T . While you can't look inside the box to see how T acts, you can *probe* the transformation by choosing various input vectors and observing what comes out.

Assume the linear transformation T is of the form $T: \mathbb{R}^n \rightarrow \mathbb{R}^m$. By probing this transformation with n vectors of a basis for the input space and observing the outputs, you can characterize the transformation T completely.

To see why this is true, consider a basis $\{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n\}$ for the n -dimensional input space $V = \mathbb{R}^n$. To characterize T , input each of the n basis vectors \vec{e}_i into the black box and record the $T(\vec{e}_i)$ that comes out.

Any input vector \vec{v} can be written as a linear combination of the basis vectors:

$$\vec{v} = v_1\vec{e}_1 + v_2\vec{e}_2 + \cdots + v_n\vec{e}_n.$$

Using these observations and the linearity of T , we can predict the output of T for this vector:

$$T(\vec{v}) = v_1T(\vec{e}_1) + v_2T(\vec{e}_2) + \cdots + v_nT(\vec{e}_n).$$

This black box model of probing is used in many areas of science and is one of the most important ideas in linear algebra. The transformation T could be the description of a chemical process, an electrical circuit, or some phenomenon in biology. As long as we know that T is (or can be approximated by) a linear transformation, we can describe it completely by probing it with a small number of inputs. This is in contrast to characterizing nonlinear transformations, which correspond to arbitrarily complex input-output relationships and require significantly more probing.

Input and output spaces

Consider the linear transformation T from n -vectors to m -vectors:

$$T: \mathbb{R}^n \rightarrow \mathbb{R}^m.$$

The **domain of the function** input space of the linear transformation T is \mathbb{R}^n and its **codomain** output space is \mathbb{R}^m . The output space is also called the *target space* and it is similar to the notion of the *target set* for functions. The input space is identical to the *domain* of the linear transformation, since $T(\vec{v})$ is defined for all inputs $\vec{v} \in \mathbb{R}^n$.

The *image space* $\text{Im}(T)$ consists of all possible outputs of the transformation T . The image space of a linear transformation is a subset of

the ~~its~~ output space, $\text{Im}(T) \subseteq \mathbb{R}^m$. A linear transformation T whose image space is equal to its ~~codomain~~ output space ($\text{Im}(T) = \mathbb{R}^m$) is called ~~surjective~~ surjective or onto. ~~Recall that a function is surjective if it covers the entire output set.~~

The *kernel* of T is the subspace of the domain \mathbb{R}^n that is mapped to the zero vector by T : $\text{Ker}(T) = \{\vec{v} \in \mathbb{R}^n \mid T(\vec{v}) = \vec{0}\}$. $\text{Ker}(T) \stackrel{\text{def}}{=} \{\vec{v} \in \mathbb{R}^n \mid T(\vec{v}) = \vec{0}\}$. A linear transformation with an empty kernel $\text{Ker}(T) = \{\vec{0}\}$ is called ~~injective~~ injective. Injective transformations map different inputs to different outputs.

If a linear transformation T is both injective and surjective, it is called ~~bijective~~ bijective. In this case, T is a *one-to-one correspondence* between the input vector space and the output vector space.

Note the terminology used to characterize linear transformations (injective, surjective, and bijective) is the same as the terminology used to characterize functions in Section ?? . Indeed, we can use the same terminology since linear transformations are functions. The concepts of image space and kernel are illustrated in Figure 5.4.

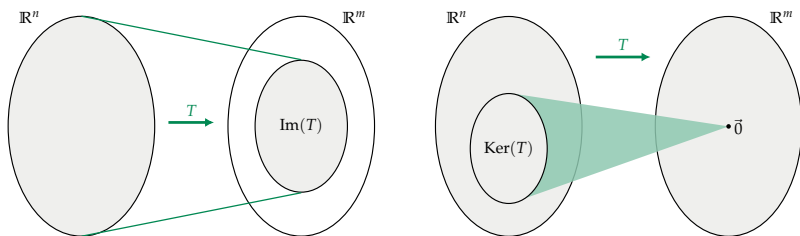


Figure 5.4: Pictorial representations of the image space $\text{Im}(T)$ and the kernel $\text{Ker}(T)$ of a linear transformation $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$. The image space is the set of all possible outputs of T . The kernel of T is the set of inputs that T maps to the zero vector.

Observation The dimensions of the input space and the output space of a bijective linear transformation must be the same. Indeed, if $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is bijective, then it is both injective and surjective. Since T is surjective, the input space must be at least as large as the output space; $n \geq m$. Since T is injective, the output space must be larger or equal to the input space; $m \geq n$. Combining these observations, we find that if $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is bijective then $m = n$.

Example 2 Consider the linear transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ defined by the equation $T(x, y, z) = (x, z)$. Find the kernel and the image space of T . Is T injective? Is T surjective?

The action of T is to delete the y -components of inputs. Any vector that has only a y -component will be sent to the zero vector. We

have $\text{Ker}(T) = \text{span}((0, 1, 0))$. The image space is $\text{Im}(T) = \mathbb{R}^2$. The transformation T is not injective. As an explicit example proving T is not injective, observe that $T(0, 1, 0) = T(0, 2, 0)$ but $(0, 1, 0) \neq (0, 2, 0)$. Since $\text{Im}(T)$ is equal to the **codomain-output space** \mathbb{R}^2 , T is surjective.

Linear transformations as matrix multiplications

An important relationship exists between linear transformations and matrices. If you fix a basis for the input vector space and a basis for the output vector space, a linear transformation $T(\vec{v}) = \vec{w}$ can be represented as matrix multiplication $M_T \vec{v} = \vec{w}$ for some matrix M_T :

$$\vec{w} = T(\vec{v}) \quad \Leftrightarrow \quad \vec{w} = M_T \vec{v}.$$

Using this equivalence, we can re-interpret several properties of matrices as properties of linear transformations. The equivalence is useful in the other direction too, since it allows us to use the language of linear transformations to talk about the properties of matrices.

The idea of representing the action of a linear transformation as a matrix **product-multiplication** is extremely important since it transforms the abstract, mathematical description of the linear transformation T into a concrete, computational one: “take the input vector \vec{v} and multiply it from the right by the matrix M_T .” Borrowing an idea from the field of software engineering, we can understand the relationship between a linear transformation and its matrix representation as analogous to the relationship between the abstract software specification of a computer program, and the program’s concrete implementation as code that runs on a computer.

Example 3 We’ll now illustrate the “linear transformation \Leftrightarrow matrix-product” equivalence with an example. Define $\Pi_{P_{xy}}$ to be the *orthogonal projection* onto the xy -plane P_{xy} . In words, the action of this projection is to zero-out the z -component of input vectors. The matrix that corresponds to this projection is

$$T\left(\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}\right) = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} \quad \Leftrightarrow \quad M_T \vec{v} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix}.$$

Finding the matrix

In order to find the matrix representation of any linear transformation $T: \mathbb{R}^n \rightarrow \mathbb{R}^m$, it is sufficient to probe T with the n vectors in the

$$\begin{array}{ll} \text{a) } T_1(x, y, z) = (x + 2y + 3z) & \text{b) } T_2(x, y) = (1, x, y) \\ \text{c) } T_3(x, y) = (x + 2y, y + x^2) & \text{d) } T_4(x, y, z) = (z, y, x) \end{array}$$

If the transformation is linear, find its matrix representation. If the transformation is nonlinear, show an example where linearity fails.

5.1 a) T_1 is linear, $M_{T_1} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$; b) T_2 is not linear, $T_2(2, 0) \neq 2T_2(1, 0)$; c) T_3 is not linear, $T_3(2, 0) \neq 2T_3(1, 0)$; d) T_4 is linear, $M_{T_4} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$.

E5.2 Consider the transformation $T(x, y, z) = (y + z, x + z, x + y)$. Find the ~~domain, codomain, input space, the output space, the~~ kernel, and ~~image space of the image of the linear transformation~~ T . Is T injective, surjective, or bijective?

5.2 ~~$\text{Dom}(T) = \mathbb{R}^3, \text{Co}(T) = \mathbb{R}^3$~~ $T: \mathbb{R}^3 \rightarrow \mathbb{R}^3$, $\text{Ker}(T) = \{\vec{0}\}$, $\text{Im}(T) = \mathbb{R}^3$. T is injective and surjective; therefore it is bijective.

E5.3 Consider the matrix $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and the linear transformations T_M and T_{M^T} defined through left and right multiplication by M : ~~$T_M(\vec{v}) = M\vec{v}$ and $T_{M^T}(\vec{u}) = \vec{u}^T M$~~ $T_M(\vec{v}) = M\vec{v}$ and $T_{M^T}(\vec{u}) = \vec{u}^T M$. Find the ~~domain, codomain, input, output,~~ kernel, and image spaces of the ~~linear~~ transformations T_M and T_{M^T} .

5.3 ~~$\text{Dom}(T_M) = \mathbb{R}^3, \text{Co}(T_M) = \mathbb{R}^2$~~ $T_M: \mathbb{R}^3 \rightarrow \mathbb{R}^2$, $\text{Ker}(T_M) = \text{span}((0, 1, 0))$, $\text{Im}(T_M) = \mathbb{R}^2$. ~~$\text{Dom}(T_{M^T}) = \mathbb{R}^2, \text{Co}(T_{M^T}) = \mathbb{R}^3$~~ $T_{M^T}: \mathbb{R}^2 \rightarrow \mathbb{R}^3$, $\text{Ker}(T_{M^T}) = \{\vec{0}\}$, $\text{Im}(T_{M^T}) = \text{span}((1, 0, 0), (0, 0, 1))$.

E5.4 What linear transformation $T: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ takes the vector $(1, 0)$ to the vector $(1, 2, 3)$, and the vector $(0, 1)$ to the vector $(4, 5, 6)$? Express your answer as a function $T(x, y) = \dots$ and as a matrix M_T .

5.4 $T(x, y) = (x + 4y, 2x + 5y, 3x + 6y)$ and $M_T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$.

5.2 Finding matrix representations

Every linear transformation $T: \mathbb{R}^n \rightarrow \mathbb{R}^m$ can be represented as a matrix $M_T \in \mathbb{R}^{m \times n}$. Suppose you're given the following description of a linear transformation: " T is the counterclockwise rotation of all points in the xy -plane by 30° ," and you want to find the matrix M_T that corresponds to this transformation.

Do you know how to find the matrix representation of T ? This section describes a simple and intuitive probing procedure for finding matrix representations. Don't worry; no alien technology is involved, and we won't be probing any humans—only linear transformations! As you read, try to bridge your understanding between the

Matrix components

Matrix entries

Every linear transformation T can be represented as a matrix M_T . Consider the linear transformation $T : V \rightarrow W$. Assume the input vector space V is n -dimensional and let $B_V = \{\hat{e}_1, \hat{e}_2, \dots, \hat{e}_n\}$ be a basis for V . Assume the output space W is m -dimensional and let $B_W = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_m\}$ be a basis for output space of T . The **coefficients entries** of the matrix $M_T \in \mathbb{R}^{m \times n}$ depend on the bases B_V and B_W . We'll now analyze this dependence in detail.

To compute the matrix representation of T with respect to the input basis $B_V = \{\hat{e}_1, \hat{e}_2, \dots, \hat{e}_n\}$, we probe T with each of the vectors in the basis and record the outputs as the columns of a matrix:

$$[M_T]_{B_V} = \left[\begin{array}{c|c|c|c} | & | & & | \\ T(\hat{e}_1) & T(\hat{e}_2) & \cdots & T(\hat{e}_n) \\ | & | & & | \end{array} \right]_{B_V}.$$

The subscript B_V indicates the columns are built from outputs of the basis B_V . We can use the matrix $[M_T]_{B_V}$ to compute $T(\vec{v})$ for a vector \vec{v} expressed in the basis B_V : $\vec{v} = (v_1, v_2, \dots, v_n)^T_{B_V}$. The matrix-vector product produces the correct linear combination of outputs:

$$\begin{aligned} [M_T]_{B_V} [\vec{v}]_{B_V} &= \left[\begin{array}{c|c|c|c} | & | & & | \\ T(\hat{e}_1) & T(\hat{e}_2) & \cdots & T(\hat{e}_n) \\ | & | & & | \end{array} \right]_{B_V} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}_{B_V} \\ &= v_1 T(\hat{e}_1) + v_2 T(\hat{e}_2) + \cdots + v_n T(\hat{e}_n) \\ &= T(v_1 \hat{e}_1 + v_2 \hat{e}_2 + \cdots + v_n \hat{e}_n) \\ &= T(\vec{v}). \end{aligned}$$

So far we've treated the outputs of T as abstract vectors $T(\hat{e}_j) \in W$. Like all vectors in the space W , each output of T can be expressed as a vector of **coefficients components** with respect to the basis B_W . For example, the output $T(\hat{e}_1)$ can be expressed as

$$T(\hat{e}_1) = \begin{bmatrix} c_{11} \\ c_{21} \\ \vdots \\ c_{m1} \end{bmatrix}_{B_W} = c_{11} \vec{b}_1 + c_{21} \vec{b}_2 + \cdots + c_{m1} \vec{b}_m$$

for some coefficients $c_{11}, c_{21}, \dots, c_{m1}$. Similarly, the other output vectors $T(\hat{e}_j)$ can be expressed as coefficients-components with respect to the basis B_W , $T(\hat{e}_j) = (c_{1j}, c_{2j}, \dots, c_{mj})_{B_W}^T$.

We're now in a position to find the matrix representation ${}_{B_W}[M_T]_{B_V}$ of the linear transformation T , with respect to the input basis B_V and the output basis B_W :

$${}_{B_W}[M_T]_{B_V} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}_{B_V} \in \mathbb{R}^{m \times n}.$$

The action of T on a vector \vec{v} is the same as the product of ${}_{B_W}[M_T]_{B_V}$ and the vector of coefficients-components $[\vec{v}]_{B_V} = (v_1, v_2, \dots, v_n)_{B_V}^T$:

$$[T(\vec{v})]_{B_W} = {}_{B_W}[M_T]_{B_V} [\vec{v}]_{B_V}.$$

You may feel this example has stretched the limits of your attention span, but bear in mind, these nitty-gritty details hold the meaning of matrix coefficients-entries. If you can see how the *positions* of the coefficients-entries in the matrix encode the information about T and the choice of bases B_V and B_W , you're well on your way to getting it. The coefficient-entry c_{ij} in the i^{th} row and j^{th} column in the matrix ${}_{B_W}[M_T]_{B_V}$ is the i^{th} component (with respect to B_W) of the output of T when the input is \hat{e}_j .

Verify—Let's verify that the matrix representation ${}_{B_W}[M_T]_{B_V}$ correctly predicts the output of T for the input $\vec{v} = 5\hat{e}_1 + 6\hat{e}_2 = (5, 6, 0, \dots)_{B_V}^T$. Using the linearity of T , we know the correct output is $T(\vec{v}) = T(5\hat{e}_1 + 6\hat{e}_2) = 5T(\hat{e}_1) + 6T(\hat{e}_2)$. We can verify that the matrix-vector product ${}_{B_W}[M_T]_{B_V} [\vec{v}]_{B_V}$ leads to the same answer:

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 0 \\ \vdots \end{bmatrix} = 5 \begin{bmatrix} c_{11} \\ c_{21} \\ \vdots \\ c_{m1} \end{bmatrix} + 6 \begin{bmatrix} c_{12} \\ c_{22} \\ \vdots \\ c_{m2} \end{bmatrix} = 5T(\hat{e}_1) + 6T(\hat{e}_2).$$

Change of basis for matrices

Given the matrix representation ${}_{B_W}[M_T]_{B_V}$ of the linear transformation $T : V \rightarrow W$, you're asked to find the matrix representation of T with respect to different bases B'_V and B'_W . This is the *change-of-basis* task for matrices.

We'll discuss the important special case where the input space and the output space of the linear transformation are the same. Let

$T : V \rightarrow V$ be a linear transformation, and let $B = \{\hat{e}_1, \hat{e}_2, \dots, \hat{e}_n\}$ and $B' = \{\hat{e}'_1, \hat{e}'_2, \dots, \hat{e}'_n\}$ be two bases for the vector space V .

Recall the change-of-basis matrix ${}_{B'}[\mathbb{1}]_B$ that converts vectors from B coordinates to B' coordinates, and its inverse ${}_B[\mathbb{1}]_{B'}$, which converts vectors from B' coordinates to B coordinates:

$$[\vec{v}]_{B'} = {}_{B'}[\mathbb{1}]_B [\vec{v}]_B \quad \text{and} \quad [\vec{v}]_B = {}_B[\mathbb{1}]_{B'} [\vec{v}]_{B'}.$$

A clarification of notation is in order. The change-of-basis matrix ${}_{B'}[\mathbb{1}]_B$ is not equal to the identity matrix $\mathbb{1}_n$. However, the change-of-basis operation is *logically* equivalent to an identity transformation: the vector \vec{v} doesn't change—only its **coefficients** coordinates change. If you don't remember the change-of-basis operation for vectors, now's the time to flip back to Section 4.3 (page 116) and review before continuing.

Given the matrix representation ${}_B[M_T]_B$ of the linear transformation T with respect to B , we want to find the matrix ${}_{B'}[M_T]_{B'}$, which is the representation of T with respect to the basis B' . The computation is straightforward. Perform the change-of-basis operation on the input and output vectors:

$${}_{B'}[M_T]_{B'} = {}_{B'}[\mathbb{1}]_B {}_B[M_T]_B {}_B[\mathbb{1}]_{B'}.$$

This group of three matrices is interpreted as follows. Imagine an input vector $[\vec{v}]_{B'}$ multiplying the three matrices ${}_{B'}[\mathbb{1}]_B {}_B[M_T]_B {}_B[\mathbb{1}]_{B'}$ from the right. In the first step, ${}_B[\mathbb{1}]_{B'}$ converts the vector from the basis B' to the basis B so the matrix ${}_B[M_T]_B$ can be applied. In the last step, the matrix ${}_{B'}[\mathbb{1}]_B$ converts the output of ${}_B[M_T]_B$ to the basis B' . The combined effect of multiplying by this specific arrangement of three matrices is the same as applying T to the input vector \vec{v} :

$${}_{B'}[\mathbb{1}]_B {}_B[M_T]_B {}_B[\mathbb{1}]_{B'} [\vec{v}]_{B'} = [T(\vec{v})]_{B'},$$

which means

$${}_{B'}[M_T]_{B'} \equiv {}_{B'}[\mathbb{1}]_B {}_B[M_T]_B {}_B[\mathbb{1}]_{B'}.$$

This formula makes sense intuitively: to obtain a matrix with respect to a different basis, we must surround it by appropriate change-of-basis matrices.

Note the touching dimensions of the matrices are expressed with respect to the same basis. Indeed, we can think of the change-of-basis matrices as adaptors we use to express vectors in different bases. The change-of-basis operation for matrices requires two adaptors; one for the input space and one for the output space.

Proofs by contradiction

Since our arrival at the invertible matrix theorem marks an important step, we'll first quickly review some handy proof techniques, just to make sure everyone's ready. A *proof by contradiction* starts by assuming the opposite of the fact we want to prove, and after several derivation steps arrives at a contradiction—a mathematical inconsistency. Arriving at a contradiction implies our original premise is false, which means the fact we want to prove is true. Thus, to show that (A) implies (B)—denoted $(A) \Rightarrow (B)$ —we can show that not-(B) implies not-(A). ~~An example of a proof by contradiction is the proof of $\sqrt{2} \notin \mathbb{Q}$ (see page ??).~~

Review of definitions

To really make sure we're all on board before the train leaves the station, it's wise to review some definitions from previous chapters. The matrix $A \in \mathbb{R}^{n \times n}$ is *invertible* if there exists a matrix A^{-1} such that $AA^{-1} = \mathbb{I}_n = A^{-1}A$. The *null space* of A is the set of vectors that become the zero vector when multiplying A from the right: $\{\vec{v} \in \mathbb{R}^n \mid A\vec{v} = \vec{0}\}$. The *column space* $\mathcal{C}(A) \subseteq \mathbb{R}^n$ consists of all possible linear combinations of the columns of the matrix A . Similarly, the *row space* $\mathcal{R}(A) \subseteq \mathbb{R}^n$ consists of all possible linear combinations of the rows of A . The *rank* of a matrix, denoted $\text{rank}(A)$, is equal to the dimension of the row space and the column space $\text{rank}(A) = \dim(\mathcal{C}(A)) = \dim(\mathcal{R}(A))$. If any of the columns of the matrix A are linearly dependent, the determinant $|A|$ will be zero.

Now tell me, do you feel ready to board the train? Don't worry too much if some of the above definitions about matrices and vector spaces are still unclear for you. The whole point of this theorem is to solidify your understanding by demonstrating how the concepts you understand less are connected to the concepts you understand well.

Proof of the invertible matrix theorem. The moment has arrived: we'll prove the equivalence of the 10 statements in the theorem by showing a closed chain of implications between statements (1) through (7). We'll separately show the equivalences (1) \Leftrightarrow (8) \Leftrightarrow (9) and (5) \Leftrightarrow (10). Figure 5.12 shows an outline of the proof.

(1) \Rightarrow (2): Assume A is invertible so there exists an inverse matrix A^{-1} such that $A^{-1}A = \mathbb{I}_n$. Therefore, for all $\vec{b} \in \mathbb{R}^n$, the expression $\vec{x} = A^{-1}\vec{b}$ is a solution to the equation $A\vec{x} = \vec{b}$. We must show the

Kernel and null space

The *kernel* of the linear transformation T is the same as the null space of its matrix representation M_T . Recall statement (3) of the invertible matrix theorem: a matrix A is invertible if and only if its null space contains only the zero vector $\mathcal{N}(A) = \{\vec{0}\}$. The equivalent condition for linear transformations is the zero kernel condition. A linear transformation T is invertible if and only if its kernel contains only the zero vector:

$$T \text{ is invertible} \quad \Leftrightarrow \quad \text{Ker}(T) = \{\vec{0}\}.$$

In contrast, a non-invertible linear transformation S sends all vectors $\vec{x} \in \text{Ker}(S) \neq \{\vec{0}\}$ to the zero vector $S(\vec{x}) = \vec{0}$. When this happens, there is no way to undo the action of S , since we can't determine the input \vec{x} that S sent to $\vec{0}$.

Linear transformations as functions

In Section ??, we discussed the notion of *invertibility* for functions of a real variable, $f : \mathbb{R} \rightarrow \mathbb{R}$. In particular, we used the terms *injective*, *surjective*, and *bijective* to describe how a function maps different inputs from its domain to outputs in its ~~codomain~~-target set (see page ??). Since linear transformations are functions, we can apply the general terminology for functions to describe how linear transformations map different inputs to outputs.

A linear transformation is *injective* if it maps different inputs to different outputs:

$$T(\vec{v}_1) \neq T(\vec{v}_2) \text{ for all } \vec{v}_1 \neq \vec{v}_2 \quad \Leftrightarrow \quad T \text{ is injective.}$$

A linear transformation $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is *surjective* if its image space equals its ~~codomain~~-output space:

$$\text{Im}(T) = \mathbb{R}^m \quad \Leftrightarrow \quad T \text{ is surjective.}$$

All bijective functions are invertible since each output \vec{y} in the output space corresponds to exactly one \vec{x} in the input space.

Interestingly, for a linear transformation $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ to be invertible, the presence of either the injective or surjective property is sufficient. If T is injective, it must have a $\text{Ker}(T) = \{\vec{0}\}$ so it is invertible. If T is surjective, its matrix representation M_T has rank n .

Links

[Nice writeup about the invertible matrix theorem with proofs]
<http://bit.ly/InvMatThmProofs> [InvMatThmProofs2](#)

Chapter 6

Theoretical linear algebra

Let's take a trip down memory lane: [150–170](#) pages ago, we embarked on a mind-expanding journey through the land of linear algebra. We encountered vector and matrix operations. We studied systems of linear equations, solving them with row operations. We covered miles of linear transformations and their matrix representations. With the skills you've acquired to reach this point, you're ready to delve into the abstract, theoretical aspects of linear algebra—that is, since you know all the useful stuff, you can officially move on to the cool stuff.

In math, we often use abstraction to find the commonalities between different mathematical objects. These parallels give us a deeper understanding of the mathematical structures we compare. This chapter extends what we know about the vector space \mathbb{R}^n to the realm of abstract vector spaces of vector-like mathematical objects (Section 6.3). We'll discuss linear independence, find bases, and count dimensions for these abstract vector spaces. We'll define abstract inner product operations and use them to generalize the concept of orthogonality for abstract vectors (Section 6.4). We'll explore the Gram–Schmidt orthogonalization procedure for distilling orthonormal bases from non-orthonormal bases (Section 6.5). Finally, we'll introduce vectors and matrices with complex [coefficients](#) (Section 6.7). [This section—Section 6.7](#) also reviews everything we've learned in this book, so be sure to read it even if complex numbers are not required for your course. Along the way, we'll develop a taxonomy for the different types of matrices according to their properties and applications (Section 6.2). We'll also investigate matrix decompositions—techniques for splitting matrices into products of simpler matrices (Section 6.6). The chapter begins by discussing the most important decomposition technique of them all: the *eigendecomposition*, which is a way to uncover the “natural basis”

Examples

Matrices, polynomials, and functions are vector-like math objects. The following examples demonstrate how we can treat these math objects as abstract vector spaces $(V, F, +, \cdot)$.

Matrices

Consider the vector space of $m \times n$ matrices over the real numbers $\mathbb{R}^{m \times n}$. The addition operation for two matrices $A, B \in \mathbb{R}^{m \times n}$ is the usual rule of matrix addition: $(A + B)_{ij} = a_{ij} + b_{ij}$.

This vector space is mn -dimensional, which can be seen by constructing a basis for the space. The standard basis for $\mathbb{R}^{m \times n}$ consists of mn matrices with zero entries everywhere except for a single 1 in the i^{th} row and the j^{th} column. Any matrix $A \in \mathbb{R}^{m \times n}$ can be written as a linear combination of the matrices in the standard basis.

Example The standard basis B_s for the vector space $\mathbb{R}^{2 \times 2}$ is

$$\mathbf{e}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \mathbf{e}_4 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Any matrix $A \in \mathbb{R}^{2 \times 2}$ can be written as a linear combination of the elements of B_s :

$$\begin{aligned} A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} &= a_{11} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + a_{12} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + a_{21} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + a_{22} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= a_{11}\mathbf{e}_1 + a_{12}\mathbf{e}_2 + a_{21}\mathbf{e}_3 + a_{22}\mathbf{e}_4. \end{aligned}$$

In other words, A can be expressed as a vector of coefficients with respect to the basis B_s : $A = (a_{11}, a_{12}, a_{21}, a_{22})_{B_s}$.

The abstract concept of a matrix $A \in \mathbb{R}^{2 \times 2}$ can be expressed as two equivalent representations. We can think of A either as an array of coefficients with two columns and two rows, or as a four-dimensional vector of coefficients with respect to the basis B_s :

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \equiv (a_{11}, a_{12}, a_{21}, a_{22})_{B_s}.$$

We've arrived at a major *knowledge buzz* milestone: **matrices are vectors!** In precise mathematical terms, we just demonstrated the existence of an *isomorphism* between the set of 2×2 matrices and the set of four-dimensional vectors. We can add, subtract, and scale 2×2 matrices in their \mathbb{R}^4 representations. In the following exercises, we'll see how to compute the matrix trace operation $\text{Tr}(A)$ in terms of the vector representation.

Symmetric 2x2 matrices

Define the vector space consisting of 2×2 symmetric matrices

$$S(2,2) \stackrel{\text{def}}{=} \{A \in \mathbb{R}^{2 \times 2} \mid A = A^T\}$$

in combination with the usual matrix addition and scalar multiplication operations. We obtain an explicit basis for this space as follows:

$$\mathbf{v}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Any element of the vector space $S \in S(2,2)$ can be written as a linear combination of the basis **elements**: vectors:

$$\begin{aligned} S &= \begin{bmatrix} a & b \\ b & c \end{bmatrix} = a \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + c \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3. \end{aligned}$$

Since there are three vectors in a basis for $S(2,2)$, the vector space $S(2,2)$ is three-dimensional.

Note how we count the dimensions in this case. The space of 2×2 matrices is four-dimensional in general, but imposing the symmetry constraint $a_{12} = a_{21}$ eliminates one parameter, so we're left with a three-dimensional space.

Polynomials of degree n

Define the vector space $P_n(t)$ of polynomials with real coefficients and degree less than or equal to n . The “vectors” in this space are polynomials of the form

$$\mathbf{p} = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n.$$

where a_0, a_1, \dots, a_n are the coefficients of the polynomial. The coefficients of the polynomial $a_0, a_1, a_2, \dots, a_n$ are the components of the vector \mathbf{p} .

The addition of vectors $\mathbf{p}, \mathbf{q} \in P_n(t)$ is performed component-wise:

$$\begin{aligned} \mathbf{p} + \mathbf{q} &= (a_0 + a_1x + \cdots + a_nx^n) + (b_0 + b_1x + \cdots + b_nx^n) \\ &= (a_0 + b_0) + (a_1 + b_1)x + \cdots + (a_n + b_n)x^n. \end{aligned}$$

Similarly, scalar multiplication acts as you'd expect:

$$\alpha \mathbf{p} = \alpha \cdot (a_0 + a_1x + \cdots + a_nx^n) = (\alpha a_0) + (\alpha a_1)x + \cdots + (\alpha a_n)x^n.$$

The space $P_n(x)$ is $(n+1)$ -dimensional since each “vector” in this space has $n+1$ **coefficients** components.

Functions

Another interesting vector space is the set of functions $f : \mathbb{R} \rightarrow \mathbb{R}$ in combination with the point-wise addition and scalar-scalar multiplication operations:

$$\mathbf{f} + \mathbf{g} = (f + g)(x) = f(x) + g(x), \quad \alpha \mathbf{f} = (\alpha f)(x) = \alpha f(x).$$

The space of functions is *infinite*-dimensional.

Discussion

We've talked about bases, components, and dimensions of *abstract* vector spaces. Indeed, these notions are well-defined for any vector-like object. Though this section only discussed vector spaces with real coefficients-components, we can apply the same techniques to vectors with coefficients-components from any *field*. The notion of a *field* describes any number-like object for which the operations of addition, subtraction, multiplication, and division are defined. An example of another field is the set of complex numbers \mathbb{C} . We'll discuss the linear algebra of vectors and matrices with complex coefficients with complex components in Section 6.7.

In the next section, we'll define an *abstract inner product* operation and use this definition to discuss concepts like orthogonality, length, and distance in abstract vector spaces.

Links

[Further discussion and examples on Wikipedia]

http://en.wikipedia.org/wiki/Vector_space

[Examples of vector spaces]

http://wikibooks.org/wiki/Linear_Algebra/Definition_and_Examples_of_Vector_Spaces

[Abstract vector spaces explained by 3Blue1Brown]

<https://youtube.com/watch?v=TgKwz5Ikpc8>

Exercises

E6.11 Find a basis for the space of 2×2 upper triangular matrices.

6.11 $\{\mathbf{e}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\}.$

6.11 First of all we must determine dimensionality of the vector space in question. The general vector space of 2×2 matrices has four dimensions, but an upper triangular matrix A satisfies $a_{ij} = 0$ for all $i > j$, which corresponds to the constraint $a_{21} = 0$. The space of 2×2

- $\arg(z) = \tan^{-1}\left(\frac{b}{a}\right)$: the *phase* or *argument* of $z = a + bi$. Note that $\tan^{-1}\left(\frac{b}{a}\right)$ and $\arg(z)$ coincide only if $a \geq 0$. A correction is necessary to the output of $\tan^{-1}\left(\frac{b}{a}\right)$ when $a < 0$. The computer function `atan2(b, a)` returns the correct phase for all $z = a + bi$.

Complex vectors

A complex vector $\vec{v} \in \mathbb{C}^n$ is an array of n complex numbers:

$$\vec{v} = (v_1, v_2, \dots, v_n) \in \underline{(\mathbb{C}, \mathbb{C}, \dots, \mathbb{C})} \equiv \mathbb{C}^n.$$

Complex matrices

A complex matrix $A \in \mathbb{C}^{m \times n}$ is a table of numbers with m rows and n columns. An example of a 3×2 matrix with complex entries is

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \in \mathbb{C}^{3 \times 2}.$$

Hermitian transpose

The *Hermitian transpose* Hermitian transpose operation, denoted † , consists of the combination of the regular transpose ($A \rightarrow A^T$) and the complex conjugation of each entry in the matrix ($a_{ij} \rightarrow \overline{a_{ij}}$):

$$A^\dagger \stackrel{\text{def}}{=} (\overline{A^T}) = (\overline{A})^T.$$

Expressed in terms of the entries of the matrix a_{ij} , the Hermitian transpose corresponds to the transformation $a_{ij} \rightarrow \overline{a_{ji}}$. There are many mathematical terms that refer to this operation, including *Hermitian conjugate*, *complex transpose*, “dagger” operation, *conjugate transpose*, and *adjoint*.

The term *adjoint* is preferred by mathematicians and the notation A^* is used consistently in mathematics research papers. The dagger notation † is preferred by physicists and engineers, but shunned by mathematicians. Mathematicians prefer to stick with the star superscript because they feel they invented the concept. We use the notation † in this book because at some point the author had to make an allegiance with one of the two camps, and because the symbol † looks a bit like the transpose symbol T .

Linear transformations are functions of the form $T : \mathbb{R}^m \rightarrow \mathbb{R}^n$. We can *represent* these linear transformations as $m \times n$ matrices *with respect to* some choice of input and output bases.

These linear algebra ideas also apply to complex vectors and complex matrices. For example, a linear transformation from $T : \mathbb{C}^2 \rightarrow \mathbb{C}^2$ can be represented ~~in terms of the matrix product~~ as the matrix multiplication

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

Each linear transformation $T : \mathbb{C}^2 \rightarrow \mathbb{C}^2$ corresponds to some 2×2 matrix $\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$ with coefficients entries $\alpha, \beta, \gamma, \delta \in \mathbb{C}$.

The change from real ~~coefficients to complex coefficients~~ entires to complex entires has the effect of doubling the number of parameters required to describe the transformation. A 2×2 complex matrix has eight parameters, not four. Where are those eight parameters, you ask? Here:

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} = \begin{bmatrix} \operatorname{Re}\{\alpha\} & \operatorname{Re}\{\beta\} \\ \operatorname{Re}\{\gamma\} & \operatorname{Re}\{\delta\} \end{bmatrix} + \begin{bmatrix} \operatorname{Im}\{\alpha\} & \operatorname{Im}\{\beta\} \\ \operatorname{Im}\{\gamma\} & \operatorname{Im}\{\delta\} \end{bmatrix} i.$$

Each of the four coefficients entries of the matrix has a real part and an imaginary part, making for a total of eight parameters to pick when specifying the matrix.

Similarly, to specify a vector $\vec{v} \in \mathbb{C}^2$ you need to specify four parameters:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \operatorname{Re}\{v_1\} \\ \operatorname{Re}\{v_2\} \end{bmatrix} + \begin{bmatrix} \operatorname{Im}\{v_1\} \\ \operatorname{Im}\{v_2\} \end{bmatrix} i.$$

In practice, this doubling of dimensions doesn't play a role in calculations because we usually perform algebra steps with the complex coefficients entries and rarely split the matrices into their real and imaginary parts.

All the linear algebra techniques you've learned also work with complex numbers, as you'll see in the following examples.

Example 1: Solving systems of equations Suppose you're solving a problem that involves complex numbers and a system of two linear equations in two unknowns:

$$\begin{aligned} z_1 + 2z_2 &= 3 + i \\ 3z_1 + (9 + i)z_2 &= 6 + 2i. \end{aligned}$$

You're asked to find the values of the unknowns z_1 and z_2 .

This is not a coincidence: complex exponentials are in many ways the natural way to talk about rotations, periodic motion, and waves. If you pursue a career in math, physics, or engineering, you'll use complex numbers and Euler's equation formula on a daily basis.

Special types of matrices

We'll now define a few special types of matrices with complex coefficientsentries. These matrices are analogous to the special matrices we defined in Section 6.2, but their definitions are adapted to use the Hermitian conjugate-transpose operation \dagger .

Unitary matrices

A matrix U is *unitary* if it obeys $U^\dagger U = \mathbb{1}$. The norm of the determinant of a unitary matrix is 1, $|\det(U)| = 1$. For an $n \times n$ matrix U , the following statements are equivalent:

- U is unitary.
- The columns of U form an orthonormal basis.
- The rows of U form an orthonormal basis.
- The inverse of U is U^\dagger .

Unitary matrices are the complex analogues of orthogonal matrices. Indeed, if a unitary matrix U has real coefficientsentries, then $U^\dagger = U^T$ and we have $U^T U = \mathbb{1}$, which is the definition of an orthogonal matrix.

Hermitian matrices

A Hermitian matrix H is equal to its own Hermitian transpose:

$$H^\dagger = H \quad \Leftrightarrow \quad h_{ij} = \overline{h_{ji}}, \text{ for all } i, j.$$

Hermitian matrices are complex-number analogues of symmetric matrices.

A Hermitian matrix H can be freely moved from one side to the other in a complex inner product:

$$\langle H\vec{x}, \vec{y} \rangle = (H\vec{x})^\dagger \vec{y} = \vec{x}^\dagger H^\dagger \vec{y} = \vec{x}^\dagger (H\vec{y}) = \langle \vec{x}, H\vec{y} \rangle.$$

The eigenvalues of Hermitian matrices are real numbers.

Normal matrices

Previously, we defined the set of real normal matrices to be matrices that satisfy $A^T A = A A^T$. For matrices with complex ~~coefficients~~entries, the definition of a normal matrix uses the dagger: $A^\dagger A = A A^\dagger$.

* * *

Consulting the concept map in Figure 6.1 on page 212 will help you see the parallels between the different types of special matrices. I realize there's a lot of new terminology to absorb all at once, so don't worry about remembering everything. The main idea is to know that these special types of matrices exist—not to know *everything* about them.

Inner product for complex vectors

The complex inner product is an operation of the form

$$\langle \cdot, \cdot \rangle : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{C}.$$

The inner product $\langle \vec{u}, \vec{v} \rangle$ for real vectors is equivalent to the matrix ~~product~~multiplication between the row vector \vec{u}^T and the column vector \vec{v} . Extending the notion of inner product to work with complex vectors requires a modification to the inner product formula. The inner product for vectors $\vec{u}, \vec{v} \in \mathbb{C}^n$ is defined as

$$\langle \vec{u}, \vec{v} \rangle \equiv \sum_{i=1}^n \bar{u}_i v_i \equiv \vec{u}^\dagger \vec{v}.$$

The formula is similar to the inner product formula for real vectors, but uses the Hermitian transpose † instead of the regular transpose T . The inner product of two vectors $\vec{u}, \vec{v} \in \mathbb{C}^3$ is

$$\langle \vec{u}, \vec{v} \rangle = \bar{u}_1 v_1 + \bar{u}_2 v_2 + \bar{u}_3 v_3 = \begin{bmatrix} \bar{u}_1 & \bar{u}_2 & \bar{u}_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \vec{u}^\dagger \vec{v}.$$

This dagger thing is very important. Using the definition of the inner product with a dagger on the first ~~entry~~input ensures the complex inner product will obey the positive semidefinite criterion (see page 218). The inner product of a vector $\vec{v} \in \mathbb{C}^3$ with itself is

$$\langle \vec{v}, \vec{v} \rangle \equiv \vec{v}^\dagger \vec{v} = \begin{bmatrix} \bar{v}_1 & \bar{v}_2 & \bar{v}_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = |v_1|^2 + |v_2|^2 + |v_3|^2,$$

Properties of the Hermitian transpose operation

The Hermitian transpose obeys the following properties:

- $(A + B)^\dagger = A^\dagger + B^\dagger$
- $(AB)^\dagger = B^\dagger A^\dagger$
- $(ABC)^\dagger = C^\dagger B^\dagger A^\dagger$
- $(A^\dagger)^{-1} = (A^{-1})^\dagger$

Note these are the same properties as the regular transpose operation from Section 2.3 (see page 30).

Conjugate linearity in the first input

The complex inner product we defined is linear in the second **entry** ~~and conjugate-linear input~~ and conjugate-linear in the first entry input:

$$\begin{aligned}\langle \vec{v}, \alpha \vec{a} + \beta \vec{b} \rangle &= \alpha \langle \vec{v}, \vec{a} \rangle + \beta \langle \vec{v}, \vec{b} \rangle, \\ \langle \alpha \vec{a} + \beta \vec{b}, \vec{w} \rangle &= \bar{\alpha} \langle \vec{a}, \vec{w} \rangle + \bar{\beta} \langle \vec{b}, \vec{w} \rangle.\end{aligned}$$

Keep this in mind every time you deal with complex inner products. The complex inner product is not symmetric since it requires that the complex conjugation be performed on the first input. Remember, instead of $\langle \vec{v}, \vec{w} \rangle \neq \langle \vec{w}, \vec{v} \rangle$, we have $\langle \vec{v}, \vec{w} \rangle = \overline{\langle \vec{w}, \vec{v} \rangle}$.

The choice of complex conjugation in the first **entry input** is a matter of convention. In this text, we *defined* the inner product $\langle \cdot, \cdot \rangle$ with the † operation on the first **entry input**, which is known as the *physics convention*. Some old mathematics texts define the inner product of complex vectors using the complex conjugation on the second **entry input**, which makes the inner product linear in the first **entry input** and conjugate-linear in the second **entry input**. This convention is fine, too. The choice of convention doesn't matter, as long as one of the **entries inputs** is conjugated to ensure the inner product obeys the positive semidefinite requirement $\langle \vec{u}, \vec{u} \rangle \geq 0$.

Function inner product

In the section on inner product spaces, we discussed the notion of the vector space of all real-valued functions of a real variable $f: \mathbb{R} \rightarrow \mathbb{R}$, and defined an inner product between functions:

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x)g(x) dx.$$

Suppose we have two complex-valued functions $f(x)$ and $g(x)$:

$$f: \mathbb{R} \rightarrow \mathbb{C}, \quad g: \mathbb{R} \rightarrow \mathbb{C}.$$

Exercises

E7.6 Recall the Fibonacci sequence $a_n = (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots)$. Describe what happens to the ratio $\frac{a_{n+1}}{a_n}$ as $n \rightarrow \infty$.

Hint: Compare the size of $(\lambda_1)^n$ and $(\lambda_2)^n$ for large values of n .

$$7.6 \quad \lim_{n \rightarrow \infty} \frac{a_{n+1}}{a_n} = \lambda_1 = \varphi = \frac{1+\sqrt{5}}{2}.$$

7.6 Since $|\lambda_2| < 1$, as n increases the number λ_2^n tends to zero, and only the effects of λ_1^n remain. The ratio $\frac{a_{n+1}}{a_n}$ approaches λ_1 as $n \rightarrow \infty$.

7.6 Linear programming

In the early days of computing, computers were primarily used to solve optimization problems, so the term “programming” is often used to describe optimization problems. *Linear programming* is the study of linear optimization problems that involve linear constraints. These types of optimization problems play an important role in business: the whole point of corporations is to constantly optimize profits subject to time, energy, and legal constraints.

Many optimization problems can be expressed as *linear programs*. To solve an optimization problem is to find the optimal value, which is either the maximum or the minimum of some function, called the *objective function*. ~~A linear program is an optimization problem where~~ In a linear program, the objective function is a multivariable linear function $g(x_1, x_2, \dots, x_n)$ and the constraints on the variables of the problem are also linear. A linear program with n variables and m constraints is expressed as a maximization problem,

$$\max_{x_1, x_2, \dots, x_n} g(x_1, x_2, \dots, x_n) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n,$$

subject to m linear constraints,

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2,$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m.$$

The solution to this linear program is the vector (x_1, x_2, \dots, x_n) that makes $g(x_1, x_2, \dots, x_n)$ as large as possible, and also satisfies all the constraints. For example, the variables x_1, x_2, \dots, x_n could represent the production rates of n different products made by a company. If

The *least squares approximate solution* to this equation is chosen so as to minimize the total squared error function:

$$\min_{\vec{m}} S(\vec{m}) = \min_{\vec{m}} \|A\vec{m} - \vec{b}\|^2.$$

In other words, of all the possible approximate solutions \vec{m} , we must pick the one that makes the length of the vector $A\vec{m} - \vec{b}$ the smallest.

Finding the least squares approximate solution

There are two possible approaches for finding the least squares solution, denoted \vec{m}^* . We can either use calculus techniques to minimize the total squared error $S(\vec{m})$, or geometry techniques to find the shortest vector ($A\vec{m} - \vec{b}$).

Regardless of the approach chosen, the trick to finding the least squares approximate solution to $A\vec{m} \approx \vec{b}$ is to multiply the equation by A^T to obtain

$$A^T A \vec{m} = A^T \vec{b}.$$

The matrix $A^T A$ will be invertible if the columns of A are linearly independent, which is the case for most tall-and-skinny matrices. We can therefore solve for \vec{m} by multiplying both sides of the above equation by the matrix $(A^T A)^{-1}$:

$$\vec{m} = (A^T A)^{-1} A^T \vec{b}.$$

Indeed, this expression is the *least squares solution* to the optimization problem we set out to solve in the beginning of this section:

$$\vec{m}^* = \underset{\vec{m}}{\operatorname{argmin}} S(\vec{m}) = \underset{\vec{m}}{\operatorname{argmin}} \|A\vec{m} - \vec{b}\|^2 = (A^T A)^{-1} A^T \vec{b}.$$

We can refer to the linear model $y_{\vec{m}^*}$ as the *best fit* for the dataset $\{A, \vec{b}\}$ since it minimizes the sum of the prediction errors.

Pseudoinverse Moore–Penrose inverse ~~Together, when~~ When we combine the particular combination of A , its transpose A^T , and the inverse operation that we used to find the approximate solution \vec{m}^* , we get what's called the Moore–Penrose ~~pseudoinverse~~ pseudoinverse or pseudoinverse of the matrix A . We use the shorthand notation A^+ (not to be confused with A^\dagger), for the entire expression:

$$A^+ \equiv \stackrel{\text{def}}{=} (A^T A)^{-1} A^T.$$

Geometric interpretation

The solution to the least squares optimization problem,

$$\vec{m}^* = \underset{\vec{m}}{\operatorname{argmin}} \|A\vec{m} - \vec{b}\|^2,$$

can be understood geometrically as the search for the vector in the column space of A that is *closest* to the vector \vec{b} , as illustrated in Figure 7.5. As we vary the parameter vector \vec{m} , we obtain different vectors $A\vec{m} \in \mathcal{C}(A)$. Of all the points $A\vec{m}$ in the column space of A , the point $A\vec{m}^*$ is the closest to the point \vec{b} .

Let's define the *error vector* that corresponds to the difference between the model prediction $A\vec{m}$ and the actual value \vec{b} :

$$\vec{e} \stackrel{\text{def}}{=} A\vec{m} - \vec{b}.$$

Using the geometric intuition and looking at Figure 7.5, we see that the optimal solution $A\vec{m}^*$ occurs when the error vector is perpendicular to the column space of A . Recall the left fundamental **spaces** of the matrix A : its column space $\mathcal{C}(A)$ and its orthogonal complement, the left null space $\mathcal{N}(A^\top)$. Thus, if we want an error vector that is perpendicular to $\mathcal{C}(A)$, we must find an error vector that lies in the left null space of A : $\vec{e}^* \in \mathcal{N}(A^\top)$. Using the definition of the left null space,

$$\mathcal{N}(A^\top) \stackrel{\text{def}}{=} \{\vec{w} \in \mathbb{R}^N \mid \vec{w}^\top A = \vec{0}^\top\},$$

we obtain the following equation that defines \vec{m}^* :

$$(\vec{e}^*)^\top A = 0 \quad \Rightarrow \quad (A\vec{m}^* - \vec{b})^\top A = 0.$$

Taking the transpose of the last equation, we obtain $A^\top(A\vec{m}^* - \vec{b}) = 0$, which is equivalent to condition $A^\top A\vec{m}^* = A^\top \vec{b}$ used to find \vec{m}^* .

Using geometric intuition about vector spaces and orthogonality proves helpful in solving this complex optimization problem. Choosing \vec{e}^* orthogonal to $\mathcal{C}(A)$ leads to the shortest vector, $A\vec{m}^* - \vec{b}$, and produces the smallest total squared error, $S(\vec{m}^*) = \|A\vec{m}^* - \vec{b}\|^2$.

Affine models

We can also apply the Moore–Penrose **pseudoinverse-inverse** formula to more complicated models. A simple extension of a linear model is the *affine model* $y_{\vec{m},c}(\vec{x}) = \vec{m} \cdot \vec{x} + c$, which adds a constant

$y_a(\vec{x}) = m_0 + m_1x_1 + \cdots + m_nx_n$, or a quadratic model, $y_q(\vec{x}) = m_0 + m_1x_1 + \cdots + m_nx_n + m_{n+1}x_1x_1 + m_{n+2}x_1x_2 + \cdots$, for the dataset.

To fit an affine model, you'll need to preprocess $A \in \mathbb{R}^{N \times 1000}$ into $A' \in \mathbb{R}^{N \times 1001}$, and then find the inverse of $(A'^T A') \in \mathbb{R}^{1001 \times 1001}$. That's totally doable.

In contrast, the number of parameters in a quadratic model with $n = 1000$ dimensions is $\frac{1}{2}(1000 + 1)(1000 + 2) = 501501$. To fit a quadratic model, you'll need to preprocess the data matrix to obtain $A' \in \mathbb{R}^{N \times 501501}$, and then find the inverse of $(A'^T A') \in \mathbb{R}^{501501 \times 501501}$. That's a big matrix. You'll need one terabyte of memory just to store all the entries of the matrix $A'^T A'$, and computing its inverse will take a *really* long time.

So the choice is made; affine model it is. The affine model may be less accurate than a quadratic model, but if it works and provides value, you can use it. This is the reason why scientists, engineers, statisticians, and business folk are so crazy about building linear models, even though more advanced models are available. Linear models are a *sweet spot* for prediction applications: they have great modelling power, they're easy to implement, and they lead to computational problems that are easy to solve.

Links

[Further discussion about least squares problems on Wikipedia]

https://en.wikipedia.org/wiki/Linear_regression

[https://en.wikipedia.org/wiki/Linear_least_squares_\(mathematics\)](https://en.wikipedia.org/wiki/Linear_least_squares_(mathematics))

[More about the Moore–Penrose [pseudoinverse](https://en.wikipedia.org/wiki/Moore-Penrose_pseudoinverse) [inverse](https://en.wikipedia.org/wiki/Moore-Penrose_pseudoinverse)]

https://en.wikipedia.org/wiki/Moore-Penrose_pseudoinverse

Exercises

E7.7 You want to determine whether a coin is fair, so you toss it repeatedly and record the number of times it lands heads. On the first trial, you flip the coin eight times and obtain four heads, which is a heads-to-flips ratio of $\frac{4}{8}$. On subsequent trials you obtain heads-to-flips ratios of $\frac{9}{16}$, $\frac{13}{24}$, $\frac{17}{32}$, and $\frac{20}{40}$. Find the best-fitting linear model $h(x) = mx$ to describe the number of heads in a trial with x flips.

7.7 $h(x) = 0.52x$.

7.7 The data you're given consists of pairs (a_i, b_i) , where a_i is the total number of flips in the trial and b_i is the number of heads. We represent the data as a matrix of flip counts $A \in \mathbb{R}^{5 \times 1}$ and a vector

of heads counts $\vec{b} \in \mathbb{R}^5$. Applying the standard approach for finding the Moore–Penrose inverse, we find $(A'^T A')^{-1} = [\frac{1}{3520}]$. Next we find the least squares approximate solution to $A'\vec{m} = \vec{b}$ by computing $\vec{m} = (A'^T A')^{-1} A'^T \vec{b} = [0.52]$. The slope of the line of best fit is 0.52.

E7.8 Find the best-fitting affine model $y = b + mx$ to the (x, y) data points $(0, 3.9)$, $(1, 3.2)$, and $(2, 1.9)$. Perform all the calculations by hand.

Hint: Find the Moore–Penrose [pseudoinverse](#) [inverse](#).

7.8 $y = 4 - x$.

7.8 The dataset consists of a matrix of inputs x , $A = (0, 1, 2)^T$, and a vector of outputs y , $\vec{b} = (3.9, 3.2, 1.9)^T$. Since we're interested in fitting an affine model $y = b + mx$, we must augment the matrix A with a column of ones to obtain A' , and then compute $A'^T A'$ and its inverse:

$$A' = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad A'^T A' = \begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix} \quad \Rightarrow \quad (A'^T A')^{-1} = \begin{bmatrix} \frac{5}{6} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

We can now compute the Moore–Penrose [pseudoinverse](#) [inverse](#) A'^+ and obtain the approximate solution as follows:

$$A'^+ = (A'^T A')^{-1} A'^T = \begin{bmatrix} \frac{5}{6} & \frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \quad \Rightarrow \quad A'^+ \vec{b} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}.$$

Thus the best-fitting affine model for the dataset is $y = 4 - x$.

E7.9 Calculate the total squared error $S(m^*) = \|Am^* - \vec{b}\|^2$ of the best-fit linear model obtained in Example 1 (page 286). Use the SymPy calculation at `bit.ly/leastsq_ex1` as your starting point.

Hint: The Matrix method `.norm()` might come in handy.

7.9 $S(m^*) = 4704.63$.

E7.10 Revisit Example 2 (page 288) and find the total squared error of the best-fit affine model $S(\vec{m}'^*) = \|A\vec{m}'^* - \vec{b}\|^2$. You can start from the calculation provided here `bit.ly/leastsq_ex2` and extend it.

7.10 $S(\vec{m}'^*) = 433.54$.

If \vec{v} is an n -dimensional vector, then its representation in homogeneous coordinates \vec{V} is an $(n+1)$ -dimensional vector. The $(n+1) \times (n+1)$ matrix A contains the combined information about both the linear transformation T and the translation \vec{d} .

Homogeneous coordinates

Instead of using a triple of Cartesian coordinates to represent points $p = (x, y, z)_c \in \mathbb{R}^3$, we'll use the quadruple $P = (x, y, z, 1)_h \in \mathbb{R}^4$, which is a representation of the same point in ~~homogeneous coordinates~~.—homogeneous coordinates. Similarly, the vector $\vec{v} = (v_x, v_y, v_z)_c \in \mathbb{R}^3$ in Cartesian coordinates corresponds to the vector $\vec{V} = (v_x, v_y, v_z, 1)_h \in \mathbb{R}^4$ in homogeneous coordinates. Though there is no mathematical difference between points and vectors, we'll stick to the language of points as it is more natural for graphics problems.

An interesting property of homogeneous coordinates is that they're not unique: the vector $\vec{v} = (v_x, v_y, v_z)_c$ corresponds to a ~~whole set of points~~ set of points in homogeneous coordinates: $\vec{V} = \{(\alpha v_x, \alpha v_y, \alpha v_z, \alpha)_h\}$, for $\alpha \in \mathbb{R}$. This makes homogeneous coordinates invariant to scaling:

$$\begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix}_c \Leftrightarrow \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix}_h = \begin{bmatrix} 5a \\ 5b \\ 5c \\ 5 \end{bmatrix}_h = \begin{bmatrix} 500a \\ 500b \\ 500c \\ 500 \end{bmatrix}_h.$$

This is kind of weird, but this extra freedom to rescale vectors arbitrarily leads to many useful applications.

To convert from homogeneous coordinates $(X, Y, Z, W)_h = (a, b, c, d)_h$ to Cartesian coordinates, we divide each component by the W -component to obtain the equivalent vector $(X, Y, Z, W)_h = (\frac{a}{d}, \frac{b}{d}, \frac{c}{d}, 1)_h$, which corresponds to the point $(x, y, z)_c = (\frac{a}{d}, \frac{b}{d}, \frac{c}{d})_c \in \mathbb{R}^3$.

~~In the case when~~ When the underlying Cartesian space is two-dimensional, the point $p = (x, y)_c \in \mathbb{R}^2$ is written as $P = (X, Y, W)_h = (x, y, 1)_h$ in homogeneous coordinates. The homogeneous coordinates $(X, Y, W)_h = (a, b, d)_h$ (where $d \neq 0$) represent the point $(x, y)_c = (\frac{a}{d}, \frac{b}{d})_c \in \mathbb{R}^2$.

This conversion between homogeneous coordinates and Cartesian coordinates can also be understood geometrically, as illustrated in Figure 7.7. The point $(X, Y, W)_h$ in homogeneous coordinates corresponds to an infinite line in the three-dimensional XYW -

whether Alice sent \vec{m}_a or \vec{m}_b . We want to compare an Eve that has access to \vec{c} , with a “control Eve” that only has access to a random string \vec{r} that is completely uncorrelated with the message or the ciphertext. A cryptosystem is secure, according to the *indistinguishability under chosen-plaintext* security definition, if the Eve with access to \vec{c} is no better at distinguishing between messages \vec{m}_a and \vec{m}_b than she is at guessing randomly (which is the best that “control Eve” can do).

We won’t go into the formal details of the math, but we need to specify exactly what we mean by “with a probability greater than guessing randomly.” Control Eve has a completely random string \vec{r} , which contains no information about the message \vec{m} , so control Eve must guess randomly and her probability of success is $\frac{1}{2}$. An Eve that can ~~distinguishing the ciphertext $\vec{c}_a = \text{Enc}(\vec{m}_a)$~~ distinguish the ciphertext $\vec{c}_a = \text{Enc}(\vec{m}_a)$ from the ciphertext ~~$\vec{c}_b = \text{Enc}(\vec{m}_b)$~~ $\vec{c}_b = \text{Enc}(\vec{m}_b)$ with a probability significantly greater than $\frac{1}{2}$ is considered to have an advantage in distinguishing the ciphertext. Any such scheme is not considered secure in terms of IND-CPA. Intuitively, the IND-CPA definition of security captures the notion that Eve should learn no information about the message \vec{m} after seeing its ciphertext \vec{c} .

Sketch of security proof The one-time pad encryption system is secure according to IND-CPA because of the assumption we make about the shared, secret key \vec{k} —namely that it is generated randomly. Each bit $k_i \in \vec{k}$ takes on the binary value 1 with probability 50%, and the value 0 with probability 50%.

Eve knows the plaintext of the two possible messages m_a and m_b , but she can’t tell which message was sent from the ciphertext \vec{c} since she can’t distinguish between two equally likely alternative scenarios. In Scenario 1, Alice sent \vec{m}_a , and the secret key is \vec{k}_a , where $\vec{c} = \vec{m}_a \oplus \vec{k}_a$. In Scenario 2, Alice sent \vec{m}_b , and the secret key is \vec{k}_b , where $\vec{c} = \vec{m}_b \oplus \vec{k}_b$. The XOR operation combines the randomness of \vec{m} with the randomness in the secret key \vec{k} , so trying to distinguish whether \vec{m}_a or \vec{m}_b was sent is just as difficult as distinguishing \vec{k}_a and \vec{k}_b . Since \vec{k} is completely random, Eve is forced to guess randomly. Thus the probability of determining the correct message is no better than guessing, which is precisely the requirement for the definition of security.

The randomness of the shared secret key is crucial to the security of the one-time pad encryption scheme. In general, we can think of *shared randomness* (shared secret key) as a communication resource that allows for mathematically-secure private communication between two parties. But what if Alice and Bob don’t have access to shared randomness (that is, to some shared secret)? In the next

Encryption

Bob uses Alice's public encryption key whenever he wants to send Alice a secret message. To encrypt message \vec{m} , Bob uses the function Enc and Alice's public encryption key $e_{\vec{k}}$ as follows:

$$\vec{c} = \text{Enc}(\vec{m}, e_{\vec{k}}).$$

When Alice receives the ciphertext \vec{c} , she uses her private key $d_{\vec{k}}$ (that only she knows) to decrypt the message:

$$\vec{m} = \text{EncDec}(\vec{c}, d_{\vec{k}}).$$

Observe that public-key cryptosystems are inherently many-to-one: anyone who knows Alice's public key $e_{\vec{k}}$ can create encrypted messages that only she can decode.

Digital signatures

Alice can also use her public-private key pair to broadcast one-to-many *authenticated* statements \vec{s} , meaning receivers can be sure the statements they receive were sent by Alice. The math is the same; we just use the keys in the opposite order. Alice encrypts the statement \vec{s} to produce a ciphertext,

$$\vec{c} = \text{Enc}(\vec{s}, d_{\vec{k}}),$$

then publishes the encrypted post \vec{c} to her blog or a public forum. Everyone who knows Alice's public key $e_{\vec{k}}$ can decrypt the post \vec{c} to obtain the statement \vec{s} :

$$\vec{s} = \text{EncDec}(\vec{c}, e_{\vec{k}}).$$

The interesting property here is that we can be sure the statement \vec{s} was sent by Alice, since only she controls the private key $d_{\vec{k}}$. This digital signature scheme makes it difficult for any third parties to impersonate Alice, since they don't know Alice's private key $d_{\vec{k}}$. This is the principle behind *digital signatures* used in the delivery of software updates.

We don't have space in this section to delve deeper into public-key cryptography, but we'll illustrate the main ideas through an example.

Example: ssh keys for remote logins

The secure shell protocol allows users to login to remote hosts. To login as user on the server `remotehost.com`, run the command `ssh`

Chapter 8 covers the basics of probability theory, Markov chains, and the idea behind Google's PageRank algorithm for classifying web pages. Probability theory is not directly related to linear algebra, but the applications we'll discuss make heavy use of linear algebra concepts.

The laws of quantum mechanics govern physics phenomena at the femto-scale—think individual atoms. It's a common misconception to assume quantum laws are somehow mysterious or counter-intuitive, and that perhaps they can only be understood by people with a solid math background. Chapter 9 contains a concise introduction to the principles of quantum mechanics specifically tailored to people who know linear algebra. The material is adapted from lectures the author prepared for a graduate-level introductory course, so there will be no dumbing down or oversimplification. With your background in linear algebra, you can handle the real stuff.

More linear algebra applications

You'll find the following resources useful if you want to learn more about linear algebra applications.

[Three compilations of linear algebra applications]

<http://aix1.uottawa.ca/~jkhoury/app.htm>

<https://medium.com/@jeremyjkun/633383d4153f>

<http://isitespeople.math.harvard.edu/fs/~knill/docsteaching/icb-topic1011412.filesm>

[A document that describes many applications in detail]

http://gwuwww.geverstine.com/reprints/Everstine_linearalgebra.pdf

[33 miniatures: algorithmic applications of linear algebra]

<http://kam.mff.cuni.cz/~matousek/stml-53-matousek-1.pdf>

[A book about linear algebra applications to data science]

<http://amazon.com/Data-Science-from-Scratch/dp/149190142X>

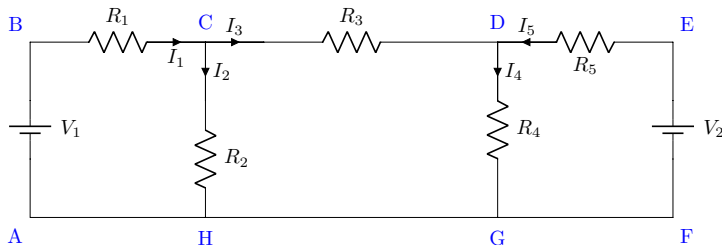
7.12 Applications problems

It would be easy to think of all the applications of linear algebra presented in this chapter as a TV program, designed to entertain rather than teach. Certainly you can continue to the next chapter without solving any problems, but do you really want to do that to yourself?

Presented next are a number of practice problems that will test your understanding of the new concepts and give you a great opportunity to practice your linear algebra skills. The linear algebra techniques we learned in previous chapters are key building blocks

for applications. So don't sit on your laurels thinking, "Yay, I'm in Chapter 7 and I know linear algebra now, I'm so good." Prove it.

P7.1 Check out this circuit containing two batteries and five resistors:



- Label the polarity of each resistor in the circuit.
- Write three KVL equations and two KCL equations.
- Rewrite the equations in the form $R\vec{I} = \vec{V}$, where R is a 5×5 matrix, $\vec{I} = (I_1, I_2, I_3, I_4, I_5)^T$, and \vec{V} is a vector of constants.
- Find the value of the currents I_1 and I_5 given $V_1 = 15[\text{V}]$, $V_2 = 10[\text{V}]$, $R_1 = 1[\Omega]$, $R_2 = 1[\Omega]$, $R_3 = 4[\Omega]$, $R_4 = 2[\Omega]$, $R_5 = 2[\Omega]$.

Hint: The direction of the voltage drop across a resistor depends on the direction of the current flowing through it.

7.1 $I_1 = -5$ $I_1 = \frac{85}{11} \approx 7.7[\text{A}]$ and $I_5 = -20$ $I_5 = \frac{25}{11} \approx 2.27[\text{A}]$. fixed

7.1 The KVL equation for the clockwise loop starting at junction B is $V_1 - R_1 I_1 - R_2 I_2 = 0$. The equation for the loop starting at C is $-R_3 I_3 - R_4 I_4 + R_2 I_2 = 0$. The equation starting at D is $+R_5 I_5 - V_2 + R_4 I_4 = 0$. The KCL equation for junction C is $I_1 = I_2 + I_3$ and that for D is $I_3 + I_5 = I_4$. After combining the equations, we obtain the matrix equation $R\vec{I} = \vec{V}$, where

$$\vec{V} = \begin{bmatrix} 15 \\ 0 \\ 10 \\ 0 \\ 0 \end{bmatrix}$$

P7.2 You just moved to a new city and you're looking for a new place to live. To get an idea of the rent prices per square foot, you check out the classifieds and find the following offers: a 200 sq ft mini-studio for \$500, a 300 sq ft studio for \$620, a 400 sq ft small apartment for \$750, a 500 sq ft one bedroom condo for \$890, a 900 sq ft two bedroom apartment for \$1250, and a 1000 sq ft apartment for \$1300. Find the best-fitting curve $p(x) = b + mx$ to the above data, where $p(x)$ represents the price for x square feet. What is the estimated price for a 700 sq ft apartment?

Hint: Use a computer algebra system like SymPy for the calculations. To get you started, I set up the problem data here: bit.ly/apt_rent_data.

7.2 $p(x) = 334 + 1.001x$; $p(700) = \$1035$.

7.2 This is a standard least squares problem with an affine model. Visit this link to see the calculations, bit.ly/apt_rent_affine; and this link to see the graph of the best-fitting affine model, bit.ly/apt_rent_fits.

The system's next state X_{t+1} depends only on its current state X_t , and not on its prior history: $X_{t-1}, X_{t-2}, \dots, X_0$. Markov chains forget everything that happened in the past, which is why we refer to them as *memoryless* random processes. A Markov chain is fully described by the conditional probability distribution $p_{X_{t+1}|X_t}(x_{t+1}|x_t)$, which describes the probability of the system's next state, given its current state. We also refer to $p_{X_{t+1}|X_t}$ as the *transition matrix*, since it describes the probabilities that the system will transition from one state to another.

Markov chains are extremely versatile models for analyzing many real-world systems. We don't have the space to cover the topic in full, but I'll introduce the basic notions for your general knowledge. First, we'll look at the connection between ~~a Markov chain and the matrix product~~ [Markov chains and matrix multiplication](#). Then, we'll see how your eigenvalue-finding skills can help compute an important property of Markov chains. Understanding Markov chains is also necessary background material for understanding Google's PageRank algorithm, which we'll discuss in Section 8.3.

Example

Three friends are kicking a football in the park: Alice, Bob, and Charlie. Whenever Alice gets the ball, she passes it to Bob 40% of the time, passes it to Charlie 40% of the time, or holds onto the ball 20% of the time. Bob is kind of a greedy dude: when he gets the ball, he holds onto it 80% of the time, and is equally likely to pass it to Alice or Charlie 20% of the time. When Charlie gets the ball, he's equally likely to pass the ball to Alice or Bob, or keep it for himself. Assume these friends kick the ball around for a very long time (hundreds of passes), and you observe them at some point. What is the probability that each player will be in possession of the ball at the instant you observe them?

We can model the ball possession as a Markov process with three possible states, $\mathcal{X} = \{A, B, C\}$, where each state describes moments when Alice, Bob, or Charlie has the ball. The transition probabilities $p_{X_{t+1}|X_t}(x_{t+1}|x_t)$ describe how the next state of the ball's possession x_{t+1} depends on the previous state of the ball's possession x_t . The transition matrix of the Markov chain in our current example is To maintain consistency with the notation for conditional probability distributions, we refer to the ~~coefficients~~ [entries](#) of M as $p_{X_{t+1}|X_t}(x_{t+1}|x_t)$. The "given" variable x_t selects the column of the matrix M , and the different entries in this column represent the transition probabilities for that state. Using the matrix M and some basic

after 10 and 20 time steps of the Markov chain:

$$p_{X_{10}} = M^{10} p_{X_0} = \begin{bmatrix} 0.161 \dots \\ 0.645 \dots \\ 0.193 \dots \end{bmatrix} \quad \text{and} \quad p_{X_{20}} = M^{20} p_{X_0} = \begin{bmatrix} 0.1612903 \dots \\ 0.6451612 \dots \\ 0.193548 \dots \end{bmatrix}.$$

Observe how the ~~coefficients~~weights of the probability distribution seem to change less and less as the Markov chain advances. Recall we originally set out to calculate the probability that each player will be in possession of the ball after a very long time. We seem to be on the right track to find this long-term probability of ball possession.

Stationary distribution

If the evolution of a Markov chain continues for long enough, the probability vector will converge to a stable distribution p_{X_∞} that remains unchanged when multiplied by M :

$$M p_{X_\infty} = p_{X_\infty}.$$

This is called the *stationary distribution* of the Markov chain. Observe that p_{X_∞} is an eigenvector of the matrix M with eigenvalue $\lambda = 1$.

The convergence to a unique stationary distribution is a fundamental property of Markov chains. Assuming the Markov chain represented by M satisfies some technical conditions (which we won't go into), it will converge to a stationary distribution p_{X_∞} . Thus, if we want to find p_{X_∞} , we just need to keep repeatedly multiplying by M until the distribution stabilizes:

$$p_{X_\infty} = \begin{bmatrix} 0.161290322580645 \dots \\ 0.645161290322581 \dots \\ 0.193548387096774 \dots \end{bmatrix} = M^\infty p_{X_0}.$$

The Markov chain will converge to the same stationary distribution p_{X_∞} regardless of the starting point p_{X_0} . The ball could start with Bob $p_{X_0} = (0, 1, 0)^T$ or with Charlie $p_{X_0} = (0, 0, 1)^T$; and after running the Markov chain for long enough, we'd still arrive at the stationary distribution p_{X_∞} .

Since we know the stationary distribution is an eigenvector of M with eigenvalue $\lambda = 1$, we can use the usual eigenvector-finding techniques to obtain p_{X_∞} directly. ~~We find the eigenvector by solving~~
~~just solve~~ for \vec{v} in $(M - \mathbb{1})\vec{v} = \vec{0}$. ~~The answer is~~ $p_{X_\infty} = \left(\frac{5}{31}, \frac{20}{31}, \frac{6}{31}\right)^T$.
~~A nice benefit of this approach is that we obtain the~~ This approach

gives us an exact analytic expression for the answer. Note we must normalize the eigenvector \vec{v} so the sum of its components equals one. The final answer is $p_{x_0} = \frac{\vec{v}}{\|\vec{v}\|_1} = (\frac{5}{31}, \frac{20}{31}, \frac{6}{31})^T$, where $\|\vec{v}\|_1 = |v_1| + |v_2| + |v_3|$ is the ℓ^1 -norm of the eigenvector.

Discussion

Markov chains have countless applications in physics, speech recognition, information processing, machine learning, and many other areas. Their simple, memoryless structure and their intuitive representation as matrices make them easy to understand and easy to fit to many situations.

In the next section, we'll describe a Markov chain model for people's web browsing behaviour. The stationary distribution of this Markov chain serves to quantify the relative importance of webpages.

Links

[Awesome visual representation of states and transitions]

<http://setosa.io/blog/2014/07/26/markov-chains/index.html>

[More details about Markov chain applications from Wikipedia]

https://en.wikipedia.org/wiki/Markov_chain

Exercises

E8.3 After reading the section on Markov chains, you decide to research the subject further by checking out a book on Markov chains from the library. In the book's notation, Markov chains are represented using *right*-multiplication of the state vector: $\vec{v}' = \vec{v}B$, where \vec{v} is the state of the system at time t , \vec{v}' is the state at time $t + 1$, and the matrix B represents the Markov chain transition probabilities.

Find the matrix B that corresponds to the transition probabilities discussed in the Markov chain example on page 359. How is this matrix B related to the matrix M that we used in the Markov chain example?

8.3 $B = M^T$.

E8.4 Go to <https://live.sympy.org> and create a Matrix object describing the passing-the-ball Markov chain transition probabilities. Use the Matrix methods `.eigenvecs()` and `.nullspace()` to confirm the stationary distribution of the Markov chain is $(\frac{5}{31}, \frac{20}{31}, \frac{6}{31})$.

Hint: You can create a Matrix object using

```
>>> M = Matrix([[ 2/10, 1/10, 1/3 ],
                 [ 4/10, 8/10, 1/3 ],
                 [ 4/10, 1/10, 1/3 ]])
```

If you start from a matrix with exact rational coefficientsentries, you'll obtain the exact answer in terms of rational numbers. You can also use the function `S` (short for `sympify`) to create rationals: $S(1/3) = \frac{1}{3}$.

Hint: The eigenspace that corresponds to the eigenvalue $\lambda = 1$ is the null space of the matrix $(M - \mathbb{I}_3)$. Use the SymPy command `eye(3)` to create a 3×3 identity matrix \mathbb{I}_3 , then apply the `nullspace` method.

Hint: Use `vec.norm(1)` to compute the ℓ^1 -norm of the vector `vec`.

$$8.4 \quad p_{X_\infty} = \left(\frac{5}{31}, \frac{20}{31}, \frac{6}{31} \right).$$

8.4 You can either use `vec = M.eigenvecs()[0][2][0]` to extract the first eigenvector, or `vec = (M-eye(3)).nullspace()[0]`, which is more efficient. To obtain a probability distribution, normalize the eigenvector by dividing it by its ℓ^1 -norm `pinf = vec/vec.norm(1)`.

8.5 Find the stationary distribution of the following Markov chain:

$$C = \begin{bmatrix} 0.8 & 0.3 & 0.2 \\ 0.1 & 0.2 & 0.6 \\ 0.1 & 0.5 & 0.2 \end{bmatrix}.$$

$$8.5 \quad p_{X_\infty} = \left(\frac{34}{61}, \frac{14}{61}, \frac{13}{61} \right).$$

8.5 Use `vec = (C-eye(3)).nullspace()[0]` to extract the eigenvector for $\lambda = 1$, then normalize the vector by its 1-norm to make it a probability distribution `pinf = vec/vec.norm(1) = \left(\frac{34}{61}, \frac{14}{61}, \frac{13}{61} \right)`.

8.3 Google's PageRank algorithm

Consider the information contained in the **links between webpages**. Each link from Page A to Page B can be interpreted as a recommendation by Page A's author for the contents of Page B. In web-speak we say links from Page A to Page B are "sending eyeballs" to Page B,

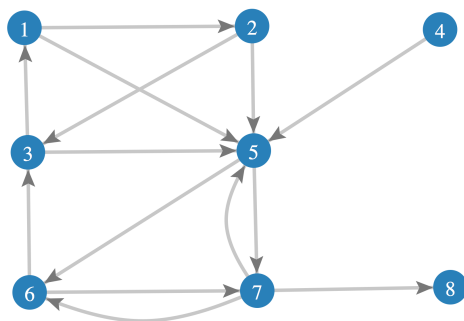


Figure 8.2: A graph showing the links between the pages on the micro-web. Page 5 seems to be an important page because many pages link to it. Since Page 5 links to pages 6 and 7, these pages will probably get a lot of eyeballs, too. Page 4 is the least important, since no links lead to it. Page 8 is an example of the unlikely case of a webpage with no outbound links.

Page 1 corresponds to the first column of the matrix M_1 . The first column of M_1 must therefore contain the coefficient-value $\frac{1}{2}$ in its second and fifth rows. Similarly, the entries in the second column of M_1 correspond to the probabilities of Randy following links randomly starting at Page 2. Since Page 2 links to Page 3 and Page 5, the second column of M_1 must contain $\frac{1}{2}$ in its third and fifth rows.

Continuing with this approach, we find the rest of the entries in the transition matrix for Strategy 1:

$$M_1 = \begin{bmatrix} 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{8} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{8} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 0 & 0 & \frac{1}{3} & \frac{1}{8} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{8} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{8} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{8} \end{bmatrix}.$$

To convince yourself that M_1 has the correct entries, compare the rest of the columns columns of M_1 with the link structure in Figure 8.2. Recall that Strategy 1 handles the exceptional case of a page with no outbound links by jumping to a random page. Since the micro-web contains eight pages, the uniform distribution over all pages is $(\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})^T$.

Do you recall the *adjacency matrix* representation for graphs we

8.7 Construct M_1 by counting the outbound links for each webpage, then mix in 0.9 of it with 0.1 of $\frac{1}{5}\mathbb{J}$ to obtain the Markov chain matrix M :

$$M_1 = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{4} & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}, \quad M_2 = \frac{1}{5}\mathbb{J}_5, \quad M = \begin{bmatrix} \frac{1}{50} & \frac{47}{100} & \frac{1}{50} & \frac{1}{50} & \frac{47}{100} \\ \frac{49}{200} & \frac{1}{50} & \frac{47}{100} & \frac{1}{50} & \frac{1}{50} \\ \frac{49}{200} & \frac{47}{100} & \frac{1}{50} & \frac{47}{100} & \frac{1}{50} \\ \frac{49}{200} & \frac{1}{50} & \frac{47}{100} & \frac{1}{50} & \frac{47}{100} \\ \frac{49}{200} & \frac{1}{50} & \frac{1}{50} & \frac{47}{100} & \frac{1}{50} \end{bmatrix}.$$

Solving $(M - \mathbb{I})\vec{e} = \vec{0}$, we find ~~$p_{X_\infty} = (0.1721, 0.169, 0.245, 0.245, 0.169)^\top$~~

$p_{X_\infty} = \frac{\vec{e}}{\|\vec{e}\|_1} = (0.1721, 0.169, 0.245, 0.245, 0.169)^\top$.

a graduate-level quantum information course, so don't think you'll be getting some watered-down, hand-wavy version of quantum mechanics. You'll learn the real stuff, because I know you can handle it.

In Section 9.6 we'll apply the quantum formalism to the polarizing lenses experiment, showing that a quantum model leads to the correct qualitative and quantitative prediction for the observed outcome. We'll close the chapter with short explanations of different applications of quantum mechanics with pointers for further exploration about each topic.

Throughout the chapter, we'll focus on *matrix* quantum mechanics and use computer science language to describe quantum phenomena. A computer science approach allows us to discuss the fundamental aspects of quantum theory without introducing all the physics required to ~~understands~~understand atoms. Finally, I just might throw in a sample calculation using the wave function of the hydrogen atom, to give you an idea of what that's like.

9.2 Polarizing lenses experiment

Let's run through a simple tabletop experiment that illustrates the limitations of classical, deterministic reasoning. The outcome of the experiment will highlight the need for careful consideration of the measurements used in scientific experiments.

We'll describe the experiment using words and diagrams, but you can easily reproduce the experiment in your own "lab," since it requires only simple equipment. I encourage you to try it yourself. You'll need three polarizing lenses, a laser pointer, a piece of paper, and three binder clips for holding the lenses upright. You can buy polarizing lenses on the cheap from a second-hand camera shop—any polarizing lens will do.

Background

In photography, polarizing lenses are used to filter out undesirable light reflections, like reflections that occur from water surfaces or glass windows. To better understand the experiment, we need to introduce some basic notions about the physics of light, specifically the concept of light polarization.

Light consists of photons. Photons are travelling pulses of electromagnetic energy. Electromagnetic energy can travel through space in the form of a wave. Polarization refers to the orientation of the electric field \vec{E} of a propagating electromagnetic wave.

Light is normally unpolarized, meaning it corresponds to a mixture of photons that have electric and magnetic components of random orientation. A light beam is *polarized* if all its photons have the same orientation of their electric field.

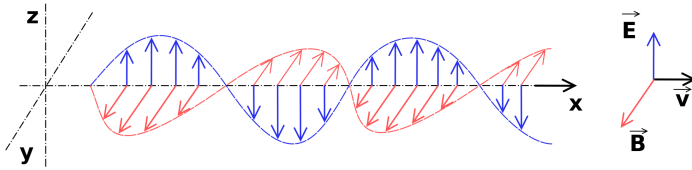


Figure 9.1: A photon is a pulse of electromagnetic energy. The energy of a photon travels in the form of a wave that has an electric component \vec{E} and a magnetic component \vec{B} . The figure shows a photon travelling in the positive x -direction with its electric component along the z -axis.

Light reflected from flat surfaces, like the surface of a lake or a glass window, becomes polarized, which means the electric components of all the reflected photons become aligned.

Photographers use this fact to selectively filter out light with a particular polarization. A *polarizing filter* or *polarizing lens* has a special coating which conducts electricity in one direction, but not in the other. You can think of a polarizing *filter* as a surface covered by tiny conductive bands that interact with the electric component of incoming light particles. Light rays that hit the *filter* will either pass through or be reflected depending on their polarization. Light particles with a polarization perpendicular to the conductive bands pass through the *filter*, while light particles with polarization parallel to the conductive bands are reflected. This is because the *filter's surface* has different conductive properties in the parallel and perpendicular directions. Note that horizontally polarizing lenses and vertically polarizing lenses are complementary: vertically polarized light will not pass through a horizontally polarizing lens. This situation is illustrated in Figure ??.

The previous examples can familiarize you with the properties of polarizing lenses, in case you don't have actual lenses to play with. If you do have polarizing lenses, you can shine a laser pointer through them, observing when light passes through *the filter* and when light is filtered out. Use binder clips to position the lenses on a flat surface, and reproduce the setup in Figure ??. Don't worry about finding the exact orientation for "vertical." Any orientation of the lenses will do, as long as the first and the second polarizing lens have the same orientation. Next, you can rotate the second lens by 90° to obtain the

quantum theory models photons as discrete packets of energy, and explains the outcome of the polarizing lenses experiment as the probability $p = \frac{1}{4}$ that a photon will pass through the circuit. For this tabletop experiment, both the classical model of light (as a wave that can be infinitely subdivided) and the quantum model of light (as discrete particles) predict the same outcome. Therefore, the polarizing lenses experiment performed with a laser pointer does not serve as proof for the necessity of a quantum mechanical description of reality.

As we discussed in the introduction to this chapter, to really reveal quantum effects, we need to look at the very small-scale or very low-power regimes. It's possible to reproduce the polarizing lenses experiment using a *single photon source*. A single photon source behaves like a super-weak laser pointer that emits only one photon at a time. When discussing the single photon regime, the classical theory of electromagnetic waves hits a wall, since a single photon cannot subdivide into parts—it is a quantum of light; a small, indivisible bundle of energy. Classical wave theory can correctly predict average optical power, but it can't provide a clear picture of what exactly happens when individual photons hit a polarizing lens.

The polarizing lenses experiment is inspired by the famous *Stern–Gerlach experiment*, which is performed with the magnetic spin of silver atoms, and which involves a similar demonstration, comparable observed outcomes, and analogous reasoning. I encourage you to learn more about the original Stern–Gerlach experiment.

[The Stern–Gerlach experiment]

https://en.wikipedia.org/wiki/Stern-Gerlach_experiment

<https://youtube.com/watch?v=rg4Fnag4V-E>

9.7 Quantum ~~physics~~ mechanics is not that weird

Without a doubt, you've heard that quantum mechanics is weird, mysterious, and generally “magical.” Well, unless vector operations count as magic, it's not *that* magical. In this section, we'll single out three so-called “weird” aspects of quantum mechanics: superposition, interference, and the fact that quantum measurements affect the states of systems being measured.

Quantum superposition

Classical binary variables (bits) can have one of two possible values: 0 or 1. Examples of physical systems that behave like bits are electric

Interference

Unlike particles that bounce off each other, waves can co-exist in the same place. The resulting wave pattern is the sum of the constituent waves. Quantum particles behave similarly to waves in certain experiments, and this can lead to *interference* between quantum systems.

The prototypical example of interference is Young's ~~double-split~~ double-slit experiment, in which particles passing through two thin slits interact with each other, causing an interference pattern of alternating bright and dark spots on a screen. Classical physics models assume particles behave like tiny point-like balls that bounce off each other whenever they come in contact. A classical model predicts that particles will appear on the screen in two bright peaks, directly facing the two slits.

In contrast, the quantum model of a particle describes it as a travelling energy pulse that exhibits wave-like properties.² In a quantum model, the particles passing through the slits behave like waves and can combine constructively or destructively, depending on the relative distances travelled by the particles. Similar interference patterns occur whenever waves combine, as in the example of waves on the surface of a liquid, or sound waves.

Performing Young's ~~double-split~~ double-slit experiment reveals a pattern of bright and dark stripes (called fringes) on the screen in support of the quantum model. The locations of the dark fringes correspond exactly to the places where particles passing through the two slits arrive "out of sync," and combine destructively:

$$|\psi\rangle - |\psi\rangle = 0.$$

The locations where destructive interference occurs correspond to the dark fringes on the screen, where no particles arrive.

The idea that one wave can cancel another wave is not new. What *is* new is the observation that particles behave like waves that can interfere with each other. That's definitely new. Indeed, interference was one of the first puzzling effects of quantum systems that was observed. Observations from interference experiments forced physicists to attribute wave-like properties to particles.

[Video demonstration of Young's ~~double-split~~ double-slit experiment]

<https://youtube.com/watch?v=qCmtegdq00A>

²This is where the name *wave function* comes from.

assume even the interactions between particles are quantized. You can think of the basic quantum mechanics described in this chapter as learning the alphabet, and quantum field theory as studying Shakespeare, including the invention of new words. Studying quantum field theory requires new math tools like path integrals, new intuitions like symmetry observations, and new computational tricks like *renormalization*.

The essential way of thinking about photons, electrons, and the interactions between them can be obtained by reading Richard Feynman's short book titled *QED*, which stands for *quantum electrodynamics*. In this tiny book, Feynman uses the analogy of a "tiny clock" attached to each particle to explain the phase $e^{i\theta}$ of a wave function. From this simple analogy, the author builds to explain complex concepts (path integrals, for instance) at the graduate level of quantum field theory. I highly recommended this book; it's a wonderful chance to learn from one of the great scientists in the field and one of the best physics teachers of all times.

[The Standard Model of particle physics]

https://en.wikipedia.org/wiki/Standard_Model

[**Nuclear fusion is how energy is generated inside stars**

https://en.wikipedia.org/wiki/Nuclear_fusion

BOOK] Richard P. Feynman. *QED: The strange theory of light and matter*. Princeton University Press, 2006, ISBN 0691125759.

Solid state physics

Physicists have sought to understand the inner structure of materials since the first days of physics. As they've learned, they've developed numerous applications, from semiconductors to lasers, photovoltaic batteries (solar panels), light emitting diodes (LEDs). These applications all depend on materials with specially engineered conductivity properties. Indeed, working with the conductivity of materials gives us insight into their other properties. We can classify materials into the following general conductivity-type groups: insulators, metals, and semi-conductors. These categories correspond to materials with different *energy band structures*.

Insulators are the most boring type of material, because their energy band structure doesn't permit any interesting chemical interactions. Take glass, for instance—just a clump of silica (SiO_2). The term *glass* is used in physics to describe any material made of randomly oriented molecules that lack a specific crystal structure.

Conductors are more interesting. A hand-wavy explanation of conductivity would be to say the electrons in conductors like aluminum and copper are "free to move around." Solid state physics

allows for a more precise understanding of the phenomenon. Using quantum mechanical models, we can determine the energy levels that electrons can occupy, and predict how many electrons will be available to conduct electricity.

Semiconductors are the most interesting type of material since they can switch between conductive and non-conductive states. The *transistor*, the invention that makes all electronics possible, consists of a sandwich of three different types of semiconductors. The voltage applied to the middle section of a transistor is called the *gate voltage*, and it controls the amount of current that can flow through the transistor. If the gate voltage is set to ON (think 1 in binary), then semiconducting material is biased; free electrons are available in its conduction band, and current can flow through. If the gate voltage is set to OFF (think 0 in binary), then the conduction band is depleted and the transistor won't conduct electricity. The improvements in semiconductor technologies—specifically the ability to pack billions of transistors into a tiny microprocessor chip—have been fuelling the ongoing computer revolution pretty much since transistors were first commercialized. In summary, no solid state physics = no mobile phones.

Quantum mechanics is used so much in solid state physics that we could suitably nickname the field “applied quantum physics.”

[Simple explanation of energy band structure and conductivity]

https://wikipedia.org/wiki/Electrical_resistivity_and_conductivity
inQ

Superconductors

Certain materials exhibit surprising physical properties at very low temperatures. By low temperatures, I mean *really low*, like -272°C ; a temperature close to *absolute zero*, the temperature at which all molecules stop. You'd exhibit surprising properties too if you were placed in an environment this cold! Take for example the properties of different metals, which are generally considered to have low electric resistance. There are regular conductors like aluminum that have low resistance, and high-end conductors like silver that have even lower resistance, and then there are *superconductors* which have zero resistance. Superconductors are an example of a purely quantum phenomenon that cannot be explained by classical physics.

Some of the most iconic landmarks of modern scientific progress, like magnetic resonance imaging (MRI) machines and magnetically levitating bullet trains, are made possible by superconductor technology. Superconductors offer zero resistance to electric current, which means they can support much stronger currents than reg-

key \vec{k} in order to use the one-time pad encryption protocol. Indeed, pre-shared entangled states are a prime resource in quantum information science. The *superdense coding protocol* is another surprising application of quantum entanglement. With this protocol, Alice can communicate two bits of classical information to Bob by sending him a single qubit and consuming one pre-shared entangled state.

Links

[Quantum simulators and practical implementations]
https://en.wikipedia.org/wiki/Quantum_simulator

[Some data about the difficulty of RSA factoring]
https://en.wikipedia.org/wiki/RSA_numbers

[~~An introduction to quantum computing~~ <http://arxiv.org/abs/0708.0261v1>]
 Video tutorials on quantum computing by Michael Nielsen]
<http://michaelnielsen.org/blog/quantum-computing-for-the-determined/>

[Grover's algorithm for unstructured search]
https://en.wikipedia.org/wiki/Grover's_algorithm

[Shor's algorithm for factoring products of prime integers]
https://en.wikipedia.org/wiki/Shor's_algorithm

[Emerging insights on limitations of quantum computing]
<https://www.archive.siam.org/pdf/news/100.pdf>

Quantum error-correcting codes

Quantum states are finicky things. Every interaction of a qubit with its environment corrupts the quantum information the qubit stores. In the previous section we talked about quantum computing in the abstract, assuming the existence of an ideal noiseless quantum computer. Since the real world is a noisy place, constructing a practical quantum computer is a much greater challenge.

Recall that errors caused by noise are also a problem for classical computers. If classical computers can be made robust to errors using error-correcting codes, can we use error-correcting codes on quantum computers too? Indeed it's possible to use *quantum error-correcting codes* to defend against the effects of quantum noise. Keep in mind, quantum error-correcting codes are more complicated to build than their classical counterparts, so it's not an obvious thing to do, but it can be done.

We won't go into too much detail, but it's worth pointing out the following interesting fact about quantum error correction. Building quantum error-correcting codes that can defend against a finite set of

Classical to Quantum Shannon Theory by Mark M. Wilde is an excellent guide to the field.

Recently, quantum theory has been applied to novel communication systems, and there is a growing interest from the communications industry to develop applications that push optical communication channels to the bounds of their theoretical efficiency. Essentially, quantum networks are being invented in parallel with quantum computers, so that when we finally build quantum computers, we'll be able to connect them together, presumably so they can share funny cat videos. What else would we use them for?

[BOOK] Mark M. Wilde. *From Classical to Quantum Shannon Theory*, Cambridge University Press, Second edition, ISBN 1107176166, <http://arxiv.org/abs/1106.1445>.

Conclusion

With this chapter, I wanted to bring you closer to the fascinating subject of quantum mechanics. I hope the material helped you understand the basic principles of quantum mechanics and clarified some of the sensational mythology surrounding the “mysteries” of the quantum world. While there's still much to discover, there's nothing too counterintuitive about quantum mechanics; it's just linear algebra, right?

One hundred years ago, quantum mechanics was seen as a foreign subject not to be trusted. In time, physicists developed good models, found better ways to explain experiments, wrote good books, and even started teaching the subject to undergraduate students. This gives me hope for humanity that we can handle even the most complex and uncertain topics when we put our minds to it.

Today we face many complex problems: consolidated corporate control of innovation, cartels, corruption, eroding democratic government systems, the militarization of everything, and conflicting ideologies. We have Sunni and Shia brothers shooting at each other and red gang versus blue gang brothers shooting at each other, and all of this for no good reason. We have all kinds of other bullshit divisions between us.

Let's hope that one hundred years from now, we'll have learned to limit the violent and corrupt aspects of human nature, so that we can realize the potential of every child born anywhere in the world. Right now it seems like it won't be an easy change, but this is how it seemed when people were trying to figure out quantum mechanics, too. All it takes is a critical mass of people who realize and truly internalize the fact we're all on the same team, and all the divisions

ponential probability distribution $p_X(x) = \lambda e^{-\lambda x}$.

To further develop your math modelling skills, the next step is to generalize the concepts of inputs x , outputs y , and functions f to other input-output relationships. In linear algebra, we studied functions of the form $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that obey the linear property:

$$T(\alpha \vec{x}_1 + \beta \vec{x}_2) = \alpha T(\vec{x}_1) + \beta T(\vec{x}_2).$$

This linear structure enables us to study the properties of many functions, solve equations involving linear transformations, and build useful models for many applications (some of which we discussed in Chapter 7). The mathematical structure of a linear transformation $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ can be ~~represented~~ represented as multiplication by a matrix $M_T \in \mathbb{R}^{m \times n}$. The notion of matrix ~~representations~~ representations ($T \Leftrightarrow M_T$) was central throughout this book. Even if you forget the computational procedures we learned, the idea of representations should stick with you, and you should be able to recognize representations in many contexts. That's a big deal, because most advanced math topics involve studying the parallels between different abstract notions. Understanding linear transformations and their ~~concrete representations as matrices~~ representations is an important ~~step in your math development~~ first step toward advanced math topics.

The computational skills you learned in Chapter 3 are also useful; though you probably won't be solving any problems by hand using row operations from this point forward, since computers outclass humans on matrix arithmetic tasks. Good riddance. Until now, you did all the work and used SymPy to check your answers. From now on, you can let SymPy do all the calculations and your job will be to chill.

If you didn't skip the sections on abstract vector spaces, you know about the parallels between the vector space \mathbb{R}^4 and the abstract vector spaces of third-degree polynomials $a_0 + a_1x + a_2x^2 + a_3x^3$ and 2×2 matrices $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$. This is another step up the ladder of abstraction, as it deepens your understanding of all math objects with vector-like structure.

It was my great pleasure to be your guide through the subject of linear algebra. I hope you walk away from this book with a solid understanding of how the concepts of linear algebra fit together. In the book's introduction, I likened linear algebra to playing with LEGOs. Indeed, if you feel comfortable manipulating vectors and matrices, performing change-of-basis operations, and using the matrix decomposition techniques to see inside matrices, you'll be able to "play" with all kinds of complex systems and problems. For example, consider the linear transformation T that you want to apply to an in-

put vector \vec{v} . Suppose the linear transformation T is most easily described in the basis B' , but the vector \vec{v} is expressed with respect to the basis B . “No problem,” you can say, and proceed to build the following chain of matrices that compute the output vector \vec{w} :

$$[\vec{w}]_B = {}_B[\mathbb{1}]_{B' \ B'} [A_T]_{B' \ B'} [{}_B[\mathbb{1}]_B [\vec{v}]_B.$$

Do you see how matrices and vectors fit together neatly like LEGOs?

I can’t tell you what the next step on your journey will be. With your new linear algebra modelling skills, a thousand doors have opened for you; now you must explore and choose. Will you learn how to code and start a software company? Maybe you’ll use your analytical skills to go to Wall Street and destroy the System from the inside. Or perhaps you’ll apply your modelling skills to revolutionize energy generation, thus making human progress sustainable. Regardless of your choice of career, I hope you’ll stay on good terms with math and continue learning whenever you have the chance. Good luck with your studies!

Social stuff

Be sure to contact me if you have any feedback about this book. It helps to hear which parts of the book readers like, hate, or don’t understand. I consider all feedback in updating and improving future editions of this book. This is how the book got good in the first place—lots of useful feedback from readers. You can reach me by email at ivan@minireference.com.

Another ~~appreciated~~ thing you can do to help us is to ~~write a review of~~ review the book on Amazon.com, Goodreads, Google Books, or otherwise spread the word about the ~~no-bullshit~~ No Bullshit Guide textbook series. Talk to your friends and let them in on the ~~math~~ knowledge buzz.

If you want to know what ~~Minireference Co.~~ Minireference Co. has been up to, check out our blog at minireference.com/blog/. The blog is a mix of 30% publishing technology talk, 50% startup business talk, and 20% announcements. Checking the blog is the easiest way to follow the progress of our revolution in the textbook industry. For real; we’re totally serious about making education accessible through affordable textbooks. You can also connect via Twitter [@minireference](https://twitter.com/minireference) and Facebook [fb.me/noBSguide](https://www.facebook.com/noBSguide).

Index

The page numbers shown in **bold** point to the concept definitions.

- absolute value, 11, **55**, 98
- abstract vector space, **318**, 348
- adjacency matrix, **373**, 460
- adjoint, **238**, 340, 354
- adjugate matrix, **196**, 208
- affine transformation, 51, 152, 385, **389**, 393
- algebraic multiplicity, **301**, 308
- anticommutative, 141, 192
- arc length, 68
- area, 14, 66, 67, 150, 188
- associative, 14, **24**, 181, 319
- augmented matrix, **165**, 214, 371
- axis, 35, 98

- basis, **94**, **226**, 241, 244, 252, 282
 - orthogonal, **226**, 304, 326
 - orthonormal, **226**, 329, 424
 - standard, 92, 225, 230, 263, 270, 300, 320, 479, 482
- Bayes' rule, 449
- Bayesian, 448
- best fit solution, 382, 384
- bijjective, **41**, 43, **262**, 293
- binary, **404**
- Born's rule, **495**
- bra, **480**, *see also* ket

- Cartesian plane, **35**, 38, 85
- Cauchy–Schwarz inequality, 325
- change of basis, 95, **229**, 230, 304, 421, 424, 482
 - for matrices, 282, 485
- characteristic polynomial, 194, **301**, 304, 345, 350
- Cholesky decomposition, **336**
- ciphertext, **403**, 407
- circle, 67
- closed under
 - addition, **233**, 234
 - scalar multiplication, **233**
- codomain, *see* output space
- coefficients, 134, 165, 226, 321
 - Fourier, 425, **427**
- cofactor, **197**, 208
- column picture, 133, 137, **146**, 290
- column space, 136, **232**, 236, 239, 246, 259, 264, 384
- column vector, 134, **147**, 238, 341
- commutative, 14, **24**, 141, 319
- completing the square, **28**, 31
- complex conjugate, **97**, 340, 480
- complex number, 13, **96**, 112
- complex transpose, *see* Hermitian transpose
- components, 76, 82, 95, **131**, 225, 323, *see also* coordinates
 - complex, 339, 480
- cone, **70**
- conjugate, **97**, 340, 480
- conjugate transpose, *see* Hermitian transpose
- conjugate-linear, **352**
- coordinate system, *see also* basis
 - Cartesian, **35**, 82, 92
 - polar, 97
- coordinates, 3, 83, 93, **131**, 210,

- 225, 245, 264, 329, 481,
see also components
- homogeneous, **390**
- cosine, 20, **62**, 72
- cosine rule, **67**
- Cramer's rule, **193**
- cross product, 83, **141**, 192, 215
- cryptography, **402**
- cylinder, **69**
- dagger, *see* Hermitian transpose
- De Moivre's formula, 102
- decryption, **403**, 406, 409
- degenerate eigenvalue, **301**, 304
- determinant, 150, 164, **185**, 188,
196, 286, 288, 301, 306
- determinant test, 191, **194**, 288
- diagonalization, **300**, 303, 308
- digital signature, **409**
- dimension, 172, 232, 239, 241,
245, 251, 265, 304, 318
- Dirac notation, **479**, 483
- direction vector, 170, **211**, 273
- distance, 216, 223, 325, 397
 - Hamming, **413**, 416
- distributive, **24**
- domain, **39**, 123, 261, 292, 421
- dot product, 83, **140**, 146, 181,
214, 219, 225, 323, 327
- eigenbasis, **298**, 302, 304, 306, 333
- eigendecomposition, 298, **301**,
305, 307, 310, 332, 376
- eigenspace, 280, 302, **304**, 307,
316, 351, 458
- eigenvalue, 190, 194, **298**, 306,
334, 339, 345, 375, 454
- eigenvector, **298**, 301, 307, 333,
339, 350, 376, 454
- encryption, **403**, 406, 409
- entries, *see* matrix entries
- error-correcting code, **412**, 528
- Euler's formula, **102**, 345, 350
- expand, **24**, **25**
- expected value, **445**
- exponent, 15
- exponential, 20, **64**
- factor, 15, **23**, 26
- factoring, 26, 34
- field, 318, **322**, 323, 339, 353, 354
- finite field, **353**, 404, 417, 468
- Fourier basis, 421, 425, 427, 431
- Fourier coefficients, 425, **427**
- Fourier series, **426**
- Fourier transformation, **421**, 429
- fraction, 13, 112
- free variable, **170**, 214, 238, 248
- Frobenius inner prod., *see* inner
prod. for matrices
- Frobenius norm, *see*
Hilbert–Schmidt norm
- function, **38**, 50
 - even, **59**
 - odd, **59**
- fundamental subspace, **232**, 236,
240, 261
 - basis, **246**
- Gauss–Jordan elimination, **168**,
174, 194, 199, 214, 238,
243, 343, 365, 369
- geometric multiplicity, **301**, 308
- golden ratio, **33**
- Gram–Schmidt
 - orthogonalization, **328**, 337
- graph, **373**, 458
- graphics processing unit, 398
- Grover's search algorithm, **525**
- Hadamard basis, **482**, 485, 498
- Hadamard gate, **494**, 522
- Hamming code, **416**, 419
- Hamming distance, **413**, 416
- Hermitian matrix, **346**
- Hermitian transpose, **340**, 480
- Hilbert–Schmidt norm, **326**, 349
- homogeneous coordinates, **390**
- homogeneous equations, 236
- idempotence, **274**
- identity matrix, 135, 149, 178,
198, **313**, 343, 393, 495
- image, **39**, 123, 136, 158, 258
- image space, 136, 158, **258**, 261,
264
- imaginary number, *see* complex
number

- infinity, 170, 209, 321, 425, 448
- information theory, 503, **529**
- injective, **41**, **262**, 293
- inner product, 143, **148**, 272, *see also* dot product
 - abstract, **322**, 329
 - complex, 341, **347**, 352, 481
 - for functions, 326, 352, 424
 - for matrices, 325, 348
- inner product space, **322**, 348, 490
- input set, *see* source set
- input space, **261**, 282, 293
- intersection, 172, 213, 390
- interval, **110**, 121
- inverse, **11**
 - function, 14, 20, 43, 136
 - matrix, 134, 136, 149, 196, 202, 287, 305, 308, 343
 - Moore–Penrose, **382**, 388
 - transformation, 5, **267**, 292
- invertibility test, **196**, 203
- invertible, 149, 185, 196, 203, 287
 - matrix, 185, 267, 287
 - transformation, **267**, 292
- isolate, 11, 20, 103

- kernel, 5, 136, 237, **258**, 262, 265, 292
- ket, **480**, 483, *see also* bra
- Kirchhoff’s laws, 369

- ℓ^1 -norm, 455
- ℓ^2 -norm, 143
- leading one, *see* pivot
- least squares approximate
 - solution, **379**
- left null space, **232**, 237, 239, 384
- length, 13, 72, 84, 98
- length preserving, *see* orthogonal matrix
- Leontief input-output model, 366
- line, 154, 173, **209**, 217, 221, 394
 - parametric equation, **211**
 - symmetric equation, **211**
- linear, 2, 56, **152**, 257, 323, 380
 - combination, 2, 134, 227, 231, 241, 320, 421, 511
 - equation, 103, 118, **153**, 164
 - independence, 194, 227, **240**, 243, 288, 301, 328
 - transformation, 3, 134, 183, 238, **257**, 269, 282, 312
- linear programming, **378**
- linear regression, **379**
- logarithm, 20, **65**
- low-rank approximation, **335**
- LU decomposition, **335**

- Markov chain, **450**, 457
- matrix, **131**, 136, 145, 263, 269
 - diagonal, 179, 302, **312**, 333
 - elementary, 184, **200**, 291, *see also* row operations
 - entries, **131**, 134, 145, 165, 185, 196, 266, **283**, 484
 - Hermitian, **346**
 - identity, *see* identity matrix
 - invariant properties, 286
 - lower triangular, **313**, 336
 - multiplication, 146, **147**, 181, 263, 389, 414, 452, 492
 - normal, 316
 - operations, 133, 145
 - orthogonal, **313**, 333, 346
 - permutation, **315**, 336
 - positive definite, 316
 - positive semidefinite, **316**, 327, 336
 - product, *see* matrix multiplication
 - projection, 148, 263, **270**, 316
 - rank, 167, 232, **240**, 286, 288
 - reflection, **274**, 280, 315, 392
 - representation, 136, **263**, 269, 282, 392, 446, 484
 - rotation, **277**, 314, 339, 392
 - symmetric, **313**, 320, 325, *see also* Hermitian matrix
 - trace, **149**, 286, 306
 - transpose, **147**, 238, 246, 382, *see also* Hermitian tr.
 - unitary, **346**, 492
 - upper triangular, **313**, 336
- matrix-vector product, **133**, 146, 177, 183, 238, 269, 381, 389, 425, 447

- metric, 327
- minor, **187**, 197
- nilpotent, **361**
- non-homogeneous, 236
- noncommutative, **178**, 182, 183
- nonnegative, 39, 55, **110**, 123
- norm, **143**, **324**, 348, 467, 496
 - Euclidian, **143**
 - Hilbert–Schmidt, **326**, 349
 - ℓ^1 -norm, 455
 - ℓ^2 -norm, 143
- normal vector, 210, **214**, 277, 401
- null space, 136, **232**, 235, 237, 239,
 - 242, 247, 259, 265, 274,
 - 288, 292, 301, 415
- nullity, **240**, 242, 293
- number line, 35
- objective function, 378, 380
- one-time pad, **406**
- one-to-one, *see* injective
- one-to-one correspondence, *see* bijective
- onto, *see* surjective
- origin, **36**, 55, 73
- orthogonal, 214, 227, 247, 265,
 - 323**, 328, 384, 415, 421
- basis, *see* basis
- complement, **219**, 221, 274,
 - 384, 415
- matrix, **313**, 333, 337, 346
- projection, 228, **270**, 392
- subspace, **220**, 331, 415
- sum, **239**, 265, 331
- orthogonalization, **328**, 330, 337
- orthonormal basis, *see* basis
- outer product, **148**, 272, 483, 484
- output set, *see* target set
- output space, **261**, 282, 293
- PageRank, **456**, 459, 463
- parabola, **53**
- parallelepiped, 185, 188
- parallelogram, 150, 185, 188
- parity check, **420**
- particular solution, 236, 237
- permutation, **315**, 336
- perp, *see* orthogonal complement
- perspective projection, 389, **394**
- pivot, **167**, 168, 174, 246, 251
- plaintext, **403**, 407
- plane, 154, 171, 173, **210**, 214, 217,
 - 222, 390, 401
 - general equation, **214**, 401
 - geometric equation, **214**
 - parametric equation, **215**
- point, 173, **209**, 216, 384, 389
- polar coordinates, 97, 207
- polarizing lens, **473**, 506
- polynomial, **56**, 101
- probability distribution, **441**, 496
 - conditional, **445**, 451
- projection, 148, 219, 263, **270**, 495
- pyramid, **70**
- QR decomposition, **337**, 338
- quadratic, 20, 26, 30, **53**, 58, 381
 - formula, 31, 57
- quantum, **465**, 506
 - bit, *see* qubit
 - cryptography, **519**
 - entanglement, **501**, 522, 527
 - information theory, 504, **529**
 - measurement, **495**
 - mechanics, **465**, 489, 510
 - operations, **492**
 - state, 467, **490**, 499, 523
 - superposition, **510**, 511
 - teleportation, **526**
 - wave function, 467, 469, **513**
- qubit, 468, 488, **491**
- radian, **68**, 72, 77
- random variable, **440**
- range, *see* image
- rank, 167, 192, 232, **240**, 286, 288
- rank–nullity theorem, 240, **242**
- rational, 13, 112
- reduced row echelon form, 164,
 - 167**, 171, 174, 194, 200,
 - 238, 244, 289
- reflection, **274**, 280, 315, 392
- relation, **49**, 52, *see also* function
- repetition code, **416**
- representation, 136, 165, 184, **263**,
 - 269, 392, 421, 446, 484

- Rivest–Shamir–Adleman (RSA) cryptosystem, 409
- roots, *see* solution set
- rotation, 277, 314, 339, 392
- row echelon form, 167, 336
- row operations, 166, 174, 184, 190, 198, 238, 244, 336
- row picture, 133, 137, 146
- row space, 232, 237, 239, 242, 244, 246, 259, 265, 414
- row vector, 147, 237, 341, 399, 415
- RREF, *see* reduced row echelon form
- RSA cryptosystem, 409
- sample space, 440
- scalar product, *see* dot product
- scale factor, 185, 207, 289
- set, 12, 38, 97, 109
 - difference, 109, 331
 - intersection, 109, 119
 - subset, 12, 109
 - union, 109, 110
- Shor’s factoring algorithm, 525
- similarity transformation, 286
- simplex algorithm, 378
- sine, 20, 60, 72
- sine rule, 67
- singular value decomposition, 333, 338, 349
- solution set, 11, 20, 31, 56, 118, 136, 172, 235, 237
- solution space, 170, 173
- source set, 39
- source space, *see* input space
- span, 227, 232, 233, 235, 288, 331
- sphere, 69
- spherical coordinates, 207, 514
- squared error, 380, 387
- stationary distribution, 454, 458
- subset, 235, 441
- substitution, 18, 59, 104
- surjective, 41, 262, 293
- SVD, 333, 338, 349
- swap (row op.), 166, 175, 190
- tangent, 63, 72
- target set, 39, 261, 292
- target space, *see* output space
- Taylor series, 154, 309
- term, 23, 56, 57
- trace, 149, 286, 306
- transition matrix, 451, 457
- transpose, 147, 238, 246, 340, 382
- trigonometric identities, 78
- undo, *see* inverse
- unit circle, 68, 73
- unit vector, 87, 227, 272, 468
- vector, 36, 80
 - direction, 170, 211, 273
 - linearly independent, *see* linear independence
 - normal, 210, 214, 277, 401
 - operations, 132, 138
 - unit, *see* unit vector
- vector product, *see* cross product
- vector space, 225, 231, 243, 328
 - abstract, 318, 348
 - inner product, 322, 348, 490
- vector subspace, 219, 231, 234
- view frustum, 400
- volume, 69, 188, 514
- wave function, 467, 469, 513
- XOR, 404