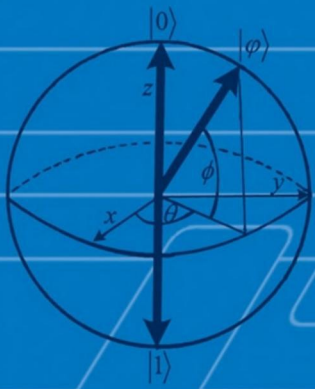
# QUANTUM COMPUTING FOR COMPUTER SCIENTISTS


Noson S. Yanofsky
Mirco A. Mannucci

CAMBRIDGE
more information – www.cambridge.org/9780521879965

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo, Delhi

Cambridge University Press
32 Avenue of the Americas, New York, NY 10013-2473, USA

www.cambridge.org
Information on this title: www.cambridge.org/9780521879965



First published 2008

Printed in the United States of America

*A catalog record for this publication is available from the British Library.*

*Library of Congress Cataloging in Publication data*

Yanofsky, Noson S., 1967–
Quantum computing for computer scientists / Noson S. Yanofsky and Mirco A. Mannucci.
p. cm.

Includes bibliographical references and index.
ISBN 978-0-521-87996-5 (hardback)
1. Quantum computers. I. Mannucci, Mirco A., 1960– II. Title.

QA76.889.Y35 2008
004.1–dc22 2008020507

ISBN 978-0-521-879965 hardback

# Contents

# Preface

Quantum computing is a fascinating new field at the intersection of computer science, mathematics, and physics, which strives to harness some of the uncanny aspects of quantum mechanics to broaden our computational horizons. This book presents some of the most exciting and interesting topics in quantum computing. Along the way, there will be some amazing facts about the universe in which we live and about the very notions of information and computation.

The text you hold in your hands has a distinct flavor from most of the other currently available books on quantum computing. First and foremost, we do not assume that our reader has much of a mathematics or physics background. This book should be readable by anyone who is in or beyond their second year in a computer science program. We have written this book specifically with computer scientists in mind, and tailored it accordingly: we assume a bare minimum of mathematical sophistication, a first course in discrete structures, and a healthy level of curiosity. Because this text was written specifically for computer people, in addition to the many exercises throughout the text, we added many programming drills. These are a hands-on, fun way of learning the material presented and getting a real feel for the subject.

The calculus-phobic reader will be happy to learn that derivatives and integrals are virtually absent from our text. Quite simply, we avoid differentiation, integration, and all higher mathematics by carefully selecting only those topics that are critical to a basic introduction to quantum computing. Because we are focusing on the fundamentals of quantum computing, we can restrict ourselves to the finite-dimensional mathematics that is required. This turns out to be not much more than manipulating vectors and matrices with complex entries. Surprisingly enough, the lion's share of quantum computing can be done without the intricacies of advanced mathematics.

Nevertheless, we hasten to stress that this is a technical textbook. We are not writing a popular science book, nor do we substitute hand waving for rigor or mathematical precision.

Most other texts in the field present a primer on quantum mechanics in all its glory. Many assume some knowledge of classical mechanics. We do not make these assumptions. We only discuss what is needed for a basic understanding of quantum computing *as a field of research in its own right*, although we cite sources for learning more about advanced topics.

There are some who consider quantum computing to be solely within the domain of physics. Others think of the subject as purely mathematical. We stress the computer science aspect of quantum computing.

It is not our intention for this book to be the definitive treatment of quantum computing. There are a few topics that we do not even touch, and there are several others that we approach briefly, not exhaustively. As of this writing, the bible of quantum computing is Nielsen and Chuang's magnificent *Quantum Computing and Quantum Information* (2000). Their book contains almost everything known about quantum computing at the time of its publication. We would like to think of our book as a useful first step that can prepare the reader for that text.

## *FEATURES*

This book is almost entirely self-contained. We do not demand that the reader come armed with a large toolbox of skills. Even the subject of complex numbers, which is taught in high school, is given a fairly comprehensive review.

The book contains many solved problems and easy-to-understand descriptions. We do not merely present the theory; rather, we explain it and go through several examples. The book also contains many exercises, which we strongly recommend the serious reader should attempt to solve. There is no substitute for rolling up one's sleeves and doing some work!

We have also incorporated plenty of programming drills throughout our text.

These are hands-on exercises that can be carried out on your laptop to gain a better understanding of the concepts presented here (they are also a great way of having fun). We hasten to point out that we are entirely language-agnostic. The student should write the programs in the language that feels most comfortable. We are also paradigm-agnostic. If declarative programming is your favorite method, go for it. If object-oriented programming is your game, use that. The programming drills build on one another. Functions created in one programming drill will be used and modified in later drills. Furthermore, in Appendix C, we show how to make little quantum computing emulators with MATLAB or how to use a ready-made one. (Our choice of MATLAB was dictated by the fact that it makes very easy-to-build, quick-and-dirty prototypes, thanks to its vast amount of built-in mathematical tools.)

This text appears to be the first to handle quantum programming languages in a significant way. Until now, there have been only research papers and a few surveys on the topic. Chapter 7 describes the basics of this expanding field: perhaps some of our readers will be inspired to contribute to quantum programming! This book also contains several appendices that are important for further study:

- Appendix A takes readers on a tour of major papers in quantum computing. This bibliographical essay was written by Jill Cirasella, Computational Sciences Specialist at the Brooklyn College Library. In addition to having a master's degree in library and information science, Jill has a master's degree in logic, for which she wrote a thesis on classical and quantum graph algorithms. This dual background uniquely qualifies her to suggest and describe further readings.
- Appendix B contains the answers to some of the exercises in the text. Other solutions will also be found on the book's Web page. We strongly urge students to do the exercises on their own and then check their answers against ours.
- Appendix C uses MATLAB, the popular mathematical environment and an established industry standard, to show how to carry out most of the mathematical operations described in this book. MATLAB has scores of routines for manipulating complex matrices: we briefly review the most useful ones and show how the reader can quickly perform a few quantum computing experiments with almost no effort, using the freely available MATLAB quantum emulator Quack.
- Appendix D, also by Jill Cirasella, describes how to use online resources to keep up with developments in quantum computing. Quantum computing is a fast-moving field, and this appendix offers guidelines and tips for finding relevant articles and announcements.
- Appendix E is a list of possible topics for student presentations. We give brief descriptions of different topics that a student might present before a class of his peers. We also provide some hints about where to start looking for materials to present.

## *ORGANIZATION*

The book begins with two chapters of mathematical preliminaries. Chapter 1 contains the basics of complex numbers, and Chapter 2 deals with complex vector spaces. Although much of Chapter 1 is currently taught in high school, we feel that a review is in order. Much of Chapter 2 will be known by students who have had a course in linear algebra. We deliberately did not relegate these chapters to an appendix at the end of the book because the mathematics is necessary to understand what is really going on. A reader who knows the material can safely skip the first two chapters. She might want to skim over these chapters and then return to them as a reference, using the index and the table of contents to find specific topics.

Chapter 3 is a gentle introduction to some of the ideas that will be encountered throughout the rest of the text. Using simple models and simple matrix multiplication, we demonstrate some of the fundamental concepts of quantum mechanics, which are then formally developed in Chapter 4. From there, Chapter 5 presents some of the basic architecture of quantum computing. Here one will find the notions of a qubit (a quantum generalization of a bit) and the quantum analog of logic gates.

Once Chapter 5 is understood, readers can safely proceed to their choice of Chapters 6 through 11. Each chapter takes its title from a typical course offered in a computer science department. The chapters look at that subfield of quantum

computing from the perspective of the given course. These chapters are almost totally independent of one another. We urge the readers to study the particular chapter that corresponds to their favorite course. Learn topics that you like first. From there proceed to other chapters.

Figure 0.1 summarizes the dependencies of the chapters.

One of the hardest topics tackled in this text is that of considering two quantum systems and combining them, or "entangled" quantum systems. This is done mathematically in Section 2.7. It is further motivated in Section 3.4 and formally presented in Section 4.5. The reader might want to look at these sections together.

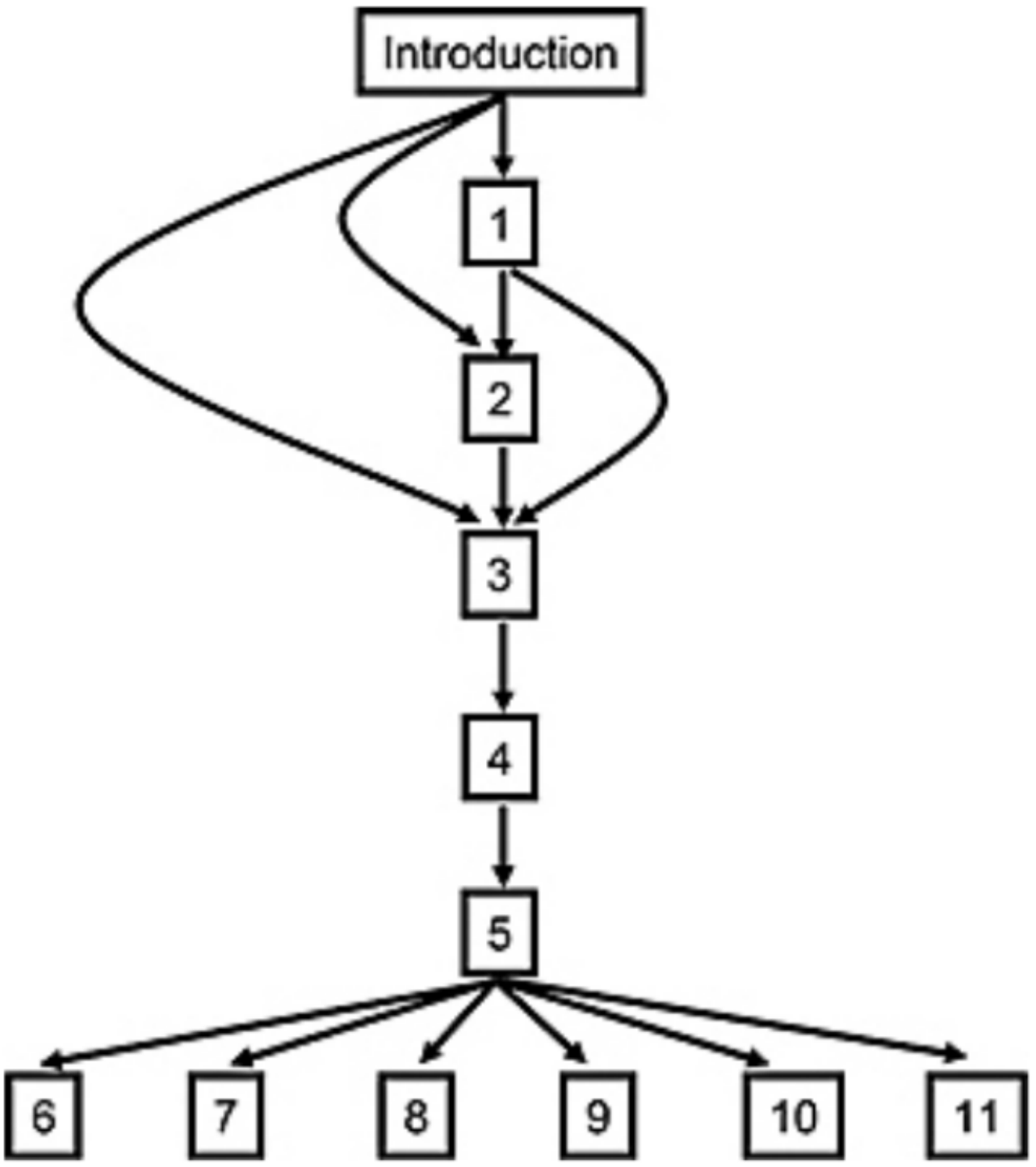



**Figure 0.1.** Chapter dependencies.

There are many ways this book can be used as a text for a course. We urge instructors to find their own way. May we humbly suggest the following three plans of action:

(1) A class that provides some depth might involve the following: Go through Chapters 1, 2, 3, 4, and 5. Armed with that background, study the entirety of Chapter 6 ("Algorithms") in depth. One can spend at least a third of a semester on that chapter. After wrestling a bit with quantum algorithms, the student will get a good feel for the entire enterprise.

(2) If breadth is preferred, pick and choose one or two sections from each of the advanced chapters. Such a course might look like this: (1), 2, 3, 4.1, 4.4, 5, 6.1, 7.1, 9.1, 10.1, 10.2, and 11. This will permit the student to see the broad outline of quantum computing and then pursue his or her own path.

(3) For a more advanced class (a class in which linear algebra and some mathematical sophistication is assumed), we recommend that students be told to read Chapters 1, 2, and 3 on their own. A nice course can then commence with Chapter 4 and plow through most of the remainder of the book.

If this is being used as a text in a classroom setting, we strongly recommend that the students make presentations. There are selected topics mentioned in Appendix E. There is no substitute for student participation!

Although we have tried to include many topics in this text, inevitably some others had to be left out. Here are a few that we omitted because of space considerations:

# Introduction

## *THE FEATURES OF THE QUANTUM WORLD*

In order to learn quantum computing, it is first necessary to become familiar with some basic facts about the quantum world. In this introduction, some unique features of quantum mechanics are introduced, as well as the way they influence the tale we are about to tell.[2]

### From Real Numbers to Complex Numbers

Quantum mechanics is different from most other branches of science in that it uses complex numbers in a fundamental way. Complex numbers were originally created as a mathematical curiosity: $i = \sqrt{-1}$ was the asserted "imaginary" solution to the polynomial equation $x^2 = -1$. As time went on, an entire mathematical edifice was constructed with these "imaginary" numbers. Complex numbers have kept lonely mathematicians busy for centuries, while physicists successfully ignored these abstract creations. However, things changed with the systematic study of wave mechanics. After the introduction of Fourier analysis, researchers learned that a compact way to represent a wave was by using functions of complex numbers. As it turns out, this was an important step on the road to using complex numbers in quantum theory. Early quantum mechanics was largely based on wave mechanics.

At first glance, we do not seem to experience complex numbers in the "real world." The length of a rod is a real number, not a complex number. The temperature outside today is 73º, not $(32 - 14i)$º. The amount of time a chemical process takes is 32.543 seconds, not $-14.65i$ seconds. One might wonder what possible role complex numbers can have in any discussion of the physical world. It will soon become apparent that they play an important, indeed an essential, role in quantum mechanics. We shall explore complex numbers in Chapters 1 and 2 of the text.

### From Single States to Superpositions of States

In order to survive in this world, human beings, as infants, must learn that every object exists in a unique place and in a well-defined state, even when we are not looking at it. Although this is true for large objects, quantum mechanics tells us that it is false for objects that are very small. A microscopic object can "hazily" be in more than one place at one time. Rather than an object's being in one position or another, we say that it is in a "superposition," i.e., in some sense, it is simultaneously in more than one location at the same time. Not only is spatial position subject to such "haziness" but so are other familiar physical properties, like energy, momentum, and certain properties that are unique to the quantum world, such as "spin."

We do not actually see superposition of states. Every time we look, or more properly, "measure," a superposition of states, it "collapses" to a single well-defined state. Nevertheless, before we measure it, it is in many states at the same time.

One is justified in greeting these claims with skepticism. After all, how can one believe something different from what every infant knows? However, we will describe certain experiments that show that this is exactly what happens.

### From Locality to Nonlocality

Central to modern science is the notion that objects are directly affected only by

nearby objects or forces. In order to determine why a phenomenon occurs at a certain place, one must examine all the phenomena and forces near[3] that place. This is called "locality," i.e., the laws of physics work in a local way. One of the most remarkable aspects of quantum mechanics is that its laws predict certain effects that work in a nonlocal manner. Two particles can be connected or "entangled" in such a way that an action performed on one of them can have an immediate effect on the other particle light-years away. This "spooky action at a distance," to use Einstein's colorful expression, was one of the most shocking discoveries of quantum mechanics.

### From Deterministic Laws to Probabilistic Laws

To which specific state will a superposition of states collapse when it is measured? Whereas in other branches of physics the laws are deterministic,[4] i.e., there is a unique outcome to every experiment, the laws of quantum mechanics state that we can only know the probability of the outcome. This, again, might seem dubious. It was doubted by the leading researchers of the time. Einstein himself was skeptical and coined the colorful expression "God does not play dice with the Universe" to express this. However, because of repeated experimental confirmations, the probabilistic nature of quantum mechanics is no longer in question.

### From Certainty to Uncertainty

The laws of quantum mechanics also inform us that there are inherent limitations to the amount of knowledge that one can ascertain about a physical system. The primary example of such a limitation is the famous "Heisenberg's uncertainty principle."

There are other important features of the quantum world that we shall not explore here. These different features were all motivating forces behind the advent of quantum computing. Rather than an historical review of how these features affected quantum computing, let us look at several areas in computer science and see how the aforementioned features affected each of those areas.[5]

## *THE IMPLICATIONS OF THE QUANTUM WORLD ON COMPUTER SCIENCE*

### Architecture

The concept of superposition will be used to generalize the notion of bit to its quantum analog, the qubit. Whereas a bit can be in either one of two states, superposition will allow a qubit to be both states simultaneously. Putting many qubits together gives us quantum registers. It is this superposition that is the basis for quantum computing's real power. Rather than being in one state at a time, a quantum computer can be in many states simultaneously.

After generalizing the notion of bit, the notion of a gate that manipulates bits will be extended to the quantum setting. We shall have quantum gates that manipulate qubits. Quantum gates will have to follow the dynamics of quantum operations. In particular, certain quantum operations are reversible, and hence certain quantum gates will have to be reversible.[6]

### Algorithms

The field of quantum algorithms uses superposition in a fundamental way. Rather than having a computer in one state at a time, one employs that aspect of the quantum world to place a quantum computer in many states simultaneously. One might think of this as massive parallelism. This needs special care: we cannot measure the computer while it is in this superposition because measuring it would collapse it to a single

position. Our algorithms will start with the quantum computer in a single position. We shall then delicately place it in a superposition of many states. From there, we manipulate the qubits in a specified way. Finally, (some of) the qubits are measured. The measurement will collapse the qubits to the desired bits, which will be our output.

Entanglement will also play a role in quantum computing, as the qubits can be entangled. By measuring some of them, others automatically reach the desired position.

Consider searching for a particular object in an unordered array. A classical algorithm examines the first entry in the array, then the second entry, and so on. The algorithm stops when either the object is found or the end of the array is reached. So for an array with $n$ elements, in the worst-case scenario, an algorithm would have to look at $n$ entries of the array.

Now imagine a computer that uses superposition. Rather than having the machine look at this entry or that entry, let it look at *all* entries simultaneously. This will result in a fantastic speedup. It turns out that such a quantum computer will be able to find the object in $\sqrt{n}$ queries to the array. This is one of the first quantum algorithms and is called “Grover’s algorithm.”

Another algorithm that demonstrates the power and usefulness of quantum computing is Shor’s algorithm for factoring numbers. The usual algorithm to factor a number involves looking at many possible factors of the number until a true factor is found. Shor’s algorithm uses superposition (and a touch of number theory) to look at many possible factors simultaneously.

Shor’s algorithm is partially based on earlier quantum algorithms that were created to solve slightly contrived problems. Although these earlier algorithms (Deutch, Deutch-Joza, and Simon’s periodicity algorithm) solve artificial problems, we shall study them so that we can learn different techniques of quantum software design.

## Programming Languages

Algorithms must eventually develop into concrete software if they are to be useful in real-life applications. The bridge that makes this step possible is programming. Quantum computing is no exception: researchers in the field have started designing quantum programming languages that will enable future generations of programmers to take control of quantum hardware and implement new quantum algorithms. We shall introduce a brief survey of programming languages (for the first time, to our knowledge, in a quantum computing textbook), starting with quantum assembler and progressing to high-level quantum programming, in particular quantum functional programming.

## Theoretical Computer Science

The goal of theoretical computer science is to formalize what engineers have done, and more important, to formalize what the engineers *cannot* do. Such an analysis is carried out by describing and classifying theoretical models of computation. The superposition of quantum mechanics has a vague feel of nondeterminism that theoretical computer scientists have used (of course, nondeterminism is a purely fictional concept and superposition is an established fact of the physical world). The indeterminacy of which state the superposition will collapse to is related to a probabilistic computation. We will be led to generalize the definition of a Turing machine to that of a quantum Turing machine. With a clear definition in place, we will be able to classify and relate all these different ideas.

We shall not only be interested in what a quantum Turing machine can do. We are also interested in the question of efficiency. This brings us to quantum complexity theory. Definitions of quantum complexity classes will be given and will be related to other well-known complexity classes.

## Cryptography

Indeterminacy and superposition will be used in quantum versions of public key distribution protocols. The fact that a measurement disturbs a quantum state shall be used to detect the presence of an eavesdropper listening in on (measuring) a communication channel. Such detection is not easily achievable in classical cryptography. Whereas classical public key distribution protocols rely on the fact that certain inverse functions are computationally hard to calculate, quantum key distribution protocols are based on the fact that certain laws of quantum physics are true. It is this strength that makes quantum cryptography so interesting and powerful.

There is also a public key protocol that uses entanglement in a fundamental way. Related to cryptography is teleportation. In teleportation, a state of a system is transported as opposed to a message. The teleportation protocol uses entangled particles that can be separated across the universe.

The most amazing part of quantum cryptography is that it is not only a theoretical curiosity. There are, in fact, actual commercially available quantum cryptography devices currently in use.

## Information Theory

It is impossible to discuss topics such as compression, transmission, and storage, without mentioning information. Information theory, now an established field, was introduced by Claude Shannon in the forties, and has developed a vast array of techniques and ideas that find their use in computer science and engineering. As this book deals with quantum computation, it is imperative that we ask: is there a satisfactory notion of quantum information? What is the information content encoded by a stream of qubits? It turns out that such notions exist. Just as classical information is related to measures of order (the so-called entropy of a source of signals), quantum information is paired with the notion of quantum entropy. We shall explore, chiefly through examples, how order and information in the quantum realm differ from familiar notions, and how these differences can be exploited to achieve new results in data storage, transmission, and compression.

## Hardware

There is no future for quantum computing without quantum computers. We are going to spell out the challenges behind the implementation of quantum machines, especially one that is embedded in the very nature of the quantum world: decoherence.We shall also describe the desirable features that a prospective quantum machine must exhibit in order to be useful.

A few proposals for quantum hardware will be showcased. The emphasis here is not on technical details (this is a book for computer scientists, not a quantum engineering handbook!). Instead, our goal is to convey the gist of these proposals and their chances of success as they are currently assessed.

[2] This Introduction is not the proper place for technical details. Some of the concepts are covered in the text and some of them can be found only in quantum mechanics textbooks. See the end of Chapter 4 for some recommendations of easy, yet detailed, introductions to quantum physics.

[3] By “near” we mean anything close enough to affect the object. In physics jargon, anything in the past light cone of the object.

[4] Statistical mechanics being one major exception.

[5] For an historical view of quantum computing as seen through the major papers that launched the subject, see Appendix A.

[6] It so happens that reversible computation has a long history predating quantum computing. This history will be reviewed in due course.

- integers (or whole numbers), $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$;
- rational numbers, $\mathbb{Q} = \left\{\frac{m}{n} | m \in \mathbb{Z}, n \in \mathbb{P}\right\}$;
- real numbers, $\mathbb{R} = \mathbb{Q} \bigcup \{\ldots, \sqrt{2}, \ldots, e, \ldots, \pi, \ldots, \frac{e}{\pi} \ldots\}$;

In none of these familiar number systems can a valid solution to Equation (1.1) be found. Mathematics often works around difficulties by simply *postulating* that such a solution, albeit unknown, is available somewhere. Let us thus boldly assume that this enigmatic solution does indeed exist and determine what it looks like: Equation (1.1) is equivalent to

$$x^2 = -1. \tag{1.2}$$

What does this state? That the solution of Equation (1.1) is a number such that its square is $-1$, i.e., a number $i$ such that

$$i^2 = -1 \quad \text{or} \quad i = \sqrt{-1}. \tag{1.3}$$

Of course we know that no such number exists among known (i.e., real) numbers, but we have already stated that this is not going to deter us. We will simply allow this new creature into the realm of well-established numbers and use it as it pleases us. Because it is *imaginary*, it is denoted $i$. We will impose on ourselves an important restriction: aside from its weird behavior when squared, $i$ will behave just like an ordinary number.

**Example 1.1.1** What is the value of $i^3$? We shall treat $i$ as a legitimate number, so

$$i^3 = i \times i \times i = (i^2) \times i = -1 \times i = -i. \tag{1.4}$$

□

**Exercise 1.1.2** Find the value of $i^{15}$. (Hint: Calculate $i$, $i^2$, $i^3$, $i^4$, and $i^5$. Find a pattern.)

■

In opening the door to our new friend $i$, we are now flooded with an entire universe of new numbers: to begin with, all the multiples of $i$ by a real number, like $2 \times i$. These fellows, being akin to $i$, are known as **imaginary numbers**. But there is more: add a real number and an imaginary number, for instance, $3 + 5 \times i$, and you get a number that is neither a real nor an imaginary. Such a number, being a hybrid entity, is rightfully called a **complex number**.

**Definition 1.1.1** *A* complex number *is an expression*

$$c = a + b \times i = a + bi, \tag{1.5}$$

*where $a$, $b$ are two real numbers; $a$ is called the real part of $c$, whereas $b$ is its imaginary part. The set of all complex numbers will be denoted as $\mathbb{C}$. When the $\times$ is understood, we shall omit it.*

Complex numbers can be added and multiplied, as shown next.

**Example 1.1.2** Let $c_1 = 3 - i$ and $c_2 = 1 + 4i$. We want to compute $c_1 + c_2$ and

$c_1 \times c_2$.

$$c_1 + c_2 = 3 - i + 1 + 4i = (3 + 1) + (-1 + 4)i = 4 + 3i. \tag{1.6}$$

Multiplying is not as easy. We must remember to multiply each term of the first complex number with each term of the second complex number. Also, remember that $i^2 = -1$.

$$\begin{aligned} c_1 \times c_2 &= (3 - i) \times (1 + 4i) = (3 \times 1) + (3 \times 4i) + (-i \times 1) + (-i \times 4i) \\ &= (3 + 4) + (-1 + 12)i = 7 + 11i. \end{aligned} \tag{1.7}$$

□

**Exercise 1.1.3** Let $c_1 = -3 + i$ and $c_2 = 2 - 4i$. Calculate $c_1 + c_2$ and $c_1 \times c_2$.

■

With addition and multiplication we can get all polynomials. We set out to find a solution for Equation (1.1); it turns out that complex numbers are enough to provide solutions for *all* polynomial equations.

**Proposition 1.1.1 (Fundamental Theorem of Algebra).** Every polynomial equation of one variable with complex coefficients has a complex solution.

**Exercise 1.1.4** Verify that the complex number $-1 + i$ is a solution for the polynomial equation $x^2 + 2x + 2 = 0$.

■

This nontrivial result shows that complex numbers are well worth our attention. In the next two sections, we explore the complex kingdom a little further.

**Programming Drill 1.1.1** *Write a program that accepts two complex numbers and outputs their sum and their product.*

## 1.2 THE ALGEBRA OF COMPLEX NUMBERS

Admittedly, the fact that we know how to handle them does not explain away the oddity of complex numbers. What *are* they? What does it mean that $i$ squared is equal to $-1$?

In the next section, we see that the geometrical viewpoint greatly aids our intuition. Meanwhile, we would like to convert complex numbers into more familiar objects by carefully looking at how they are built.

Definition 1.1.1 tells us *two* real numbers correspond to each complex number: its real and imaginary parts. A complex number is thus a two-pronged entity, carrying its two components along. How about *defining* a complex number as an ordered pair of reals?

$$c \longmapsto (a, b). \tag{1.8}$$

Ordinary real numbers can be identified with pairs $(a, 0)$

$$a \longmapsto (a, 0), \tag{1.9}$$

whereas imaginary numbers will be pairs $(0, b)$. In particular,

$$i \longmapsto (0, 1). \tag{1.10}$$

Addition is rather obvious – it adds pairs componentwise:

$$(a_1, b_1) + (a_2, b_2) = (a_1 + a_2, b_1 + b_2). \tag{1.11}$$

Multiplication is a little trickier:

$$(a_1, b_1) \times (a_2, b_2) = (a_1, b_1)(a_2, b_2) = (a_1a_2 - b_1b_2, a_1b_2 + a_2b_1). \tag{1.12}$$

Does this work? Multiplying $i$ by itself gives

$$i \times i = (0, 1) \times (0, 1) = (0 - 1, 0 + 0) = (-1, 0), \tag{1.13}$$

which is what we wanted.

Using addition and multiplication, we can write any complex number in the usual form:

$$c = (a, b) = (a, 0) + (0, b) = (a, 0) + (b, 0) \times (0, 1) = a + bi. \tag{1.14}$$

We have traded one oddity for another: $i$ was previously quite mysterious, whereas now it is just (0, 1). A complex number is nothing more than an ordered pair of ordinary real numbers. Multiplication, though, is rather strange: perhaps the reader would have expected a componentwise multiplication, just like addition. We shall see later that by viewing complex numbers through yet another looking glass the strangeness linked to their multiplication rule will fade away.

**Example 1.2.1** Let $c_1 = (3, -2)$ and $c_2 = (1, 2)$. Let us multiply them using the aforementioned rule:

$$\begin{aligned} c_1 \times c_2 &= (3 \times 1 - (-2) \times 2, -2 \times 1 + 2 \times 3) \\ &= (3 + 4, -2 + 6) = (7, 4) = 7 + 4i. \end{aligned} \tag{1.15}$$

□

**Exercise 1.2.1** Let $c_1 = (-3, -1)$ and $c_2 = (1, -2)$. Calculate their product.

■

So far, we have a set of numbers and two operations: addition and multiplication. Both operations are **commutative**, meaning that for arbitrary complex numbers $c_1$ and $c_2$,

$$c_1 + c_2 = c_2 + c_1 \tag{1.16}$$

and

$$c_1 \times c_2 = c_2 \times c_1. \tag{1.17}$$

Both operations are also **associative**:

$$(c_1 + c_2) + c_3 = c_1 + (c_2 + c_3) \tag{1.18}$$

and

$$(c_1 \times c_2) \times c_3 = c_1 \times (c_2 \times c_3). \tag{1.19}$$

**Exercise 1.2.2** Verify that multiplication of complex numbers is associative.

■

Moreover, multiplication **distributes** over addition: for all $c_1$, $c_2$, $c_3$, we have

$$c_1 \times (c_2 + c_3) = (c_1 \times c_2) + (c_1 \times c_3). \tag{1.20}$$

Let us verify this property: first we write the complex numbers as pairs $c_1 = (a_1, b_1)$, $c_2 = (a_2, b_2)$, and $c_3 = (a_3, b_3)$. Now, let us expand the left side

$$\begin{aligned} c_1 \times (c_2 + c_3) &= (a_1, b_1) \times ((a_2, b_2) + (a_3, b_3)) \\ &= (a_1, b_1) \times (a_2 + a_3, b_2 + b_3) \\ &= (a_1 \times (a_2 + a_3) - b_1 \times (b_2 + b_3), \\ &\quad a_1 \times (b_2 + b_3) + b_1 \times (a_2 + a_3)) \\ &= (a_1 \times a_2 + a_1 \times a_3 - b_1 \times b_2 - b_1 \times b_3, \\ &\quad a_1 \times b_2 + a_1 \times b_3 + b_1 \times a_2 + b_1 \times a_3). \end{aligned} \tag{1.21}$$

Turning to the right side of Equation (1.20) one piece at a time gives

$$c_1 \times c_2 = (a_1 \times a_2 - b_1 \times b_2, a_1 \times b_2 + a_2 \times b_1) \tag{1.22}$$

$$c_1 \times c_3 = (a_1 \times a_3 - b_1 \times b_3, a_1 \times b_3 + a_3 \times b_1); \tag{1.23}$$

summing them up we obtain

$$\begin{aligned} c_1 \times c_2 + c_1 \times c_3 = (&a_1 \times a_2 - b_1 \times b_2 + a_1 \times a_3 - b_1 \times b_3, \\ &a_1 \times b_2 + a_2 \times b_1 + a_1 \times b_3 + a_3 \times b_1), \end{aligned} \tag{1.24}$$

which is precisely what we got in Equation (1.21).

Having addition and multiplication, we need their complementary operations: subtraction and division.

Subtraction is straightforward:

$$c_1 - c_2 = (a_1, b_1) - (a_2, b_2) = (a_1 - a_2, b_1 - b_2); \tag{1.25}$$

in other words, subtraction is defined componentwise, as expected.

As for division, we have to work a little: If

$$(x, y) = \frac{(a_1, b_1)}{(a_2, b_2)}, \tag{1.26}$$

then by definition of division as the inverse of multiplication

$$(a_1, b_1) = (x, y) \times (a_2, b_2) \tag{1.27}$$

or

$$(a_1, b_1) = (a_2x - b_2y, a_2y + b_2x). \tag{1.28}$$

So we end up with

$$(1) \qquad a_1 = a_2x - b_2y, \tag{1.29}$$

$$(2) \qquad b_1 = a_2y + b_2x. \tag{1.30}$$

To determine the answer, we must solve this pair of equations for $x$ and $y$. Multiply both sides of (1) by $a_2$ and both sides of (2) by $b_2$. We end up with

$$(1') \qquad a_1a_2 = a_2^2x - b_2a_2y, \tag{1.31}$$

$$(2') \qquad b_1b_2 = a_2b_2y + b_2^2x. \tag{1.32}$$

Now, let us add (1′) and (2′) to get

$$a_1a_2 + b_1b_2 = (a_2^2 + b_2^2)x. \tag{1.33}$$

Solving for $x$ gives us

$$x = \frac{a_1a_2 + b_1b_2}{a_2^2 + b_2^2}. \tag{1.34}$$

We can perform the same trick for $y$ by multiplying (1) and (2) by $b_2$ and $-a_2$, respectively, and then summing. We obtain

$$y = \frac{a_2b_1 - a_1b_2}{a_2^2 + b_2^2}. \tag{1.35}$$

In more compact notation, we can express this equation as

$$\frac{a_1 + b_1i}{a_2 + b_2i} = \frac{a_1a_2 + b_1b_2}{a_2^2 + b_2^2} + \frac{a_2b_1 - a_1b_2}{a_2^2 + b_2^2}i. \tag{1.36}$$

Notice that both $x$ and $y$ are calculated using the same denominator, namely, $a_2^2 + b_2^2$. We are going to see what this quantity means presently. In the meantime, here is a concrete example.

**Example 1.2.2** Let $c_1 = -2 + i$ and $c_2 = 1 + 2i$. We will compute $\frac{c_1}{c_2}$. In this case, $a_1 = -2$, $b_1 = 1$, $a_2 = 1$, and $b_2 = 2$. Therefore,

$$a_2^2 + b_2^2 = 1^2 + 2^2 = 5, \tag{1.37}$$

$$a_1a_2 + b_1b_2 = -2 \times 1 + 1 \times 2 = 0, \tag{1.38}$$

**Exercise 1.2.11** Show that conjugation respects multiplication, i.e.,

$$\overline{c_1} \times \overline{c_2} = \overline{c_1 \times c_2}. \tag{1.47}$$

■

Notice that the function

$$c \longmapsto \overline{c} \tag{1.48}$$

given by conjugation is **bijective**, i.e., is one-to-one and onto. Indeed, two different complex numbers are never sent to the same number by conjugation. Moreover, every number is the complex conjugate of some number. A function from a field to a field that is bijective and that respects addition and multiplication is known as a **field isomorphism**. Conjugation is thus a field isomorphism of $\mathbb{C}$ to $\mathbb{C}$.

**Exercise 1.2.12** Consider the operation given by flipping the sign of the real part. Is this a field isomorphism of $\mathbb{C}$? If yes, prove it. Otherwise, show where it fails.

■

We cannot continue without mentioning another property of conjugation:

$$c \times \overline{c} = |c|^2. \tag{1.49}$$

In words, the modulus squared of a complex number is obtained by multiplying the number with its conjugate. For example,

$$(3+2i) \times (3-2i) = 3^2 + 2^2 = 13 = |3+2i|^2. \tag{1.50}$$

We have covered what we need from the algebraic perspective. We see in the next section that the geometric approach sheds some light on virtually all topics touched on here.

**Programming Drill 1.2.1** *Take the program that you wrote in the last programming drill and make it also perform subtraction and division of complex numbers. In addition, let the user enter a complex number and have the computer return its modulus and conjugate.*

## *1.3 THE GEOMETRY OF COMPLEX NUMBERS*

As far as algebra is concerned, complex numbers are an algebraically complete field, as we have described them in Section 1.2. That alone would render them invaluable as a mathematical tool. It turns out that their significance extends far beyond the algebraic domain and makes them equally useful in geometry and hence in physics. To see why this is so, we need to look at a complex number in yet another way. At the beginning of Section 1.2, we learned that a complex number is a pair of real numbers. This suggests a natural means of representation: real numbers are placed on the line, so pairs of reals correspond to points on the plane, or, equivalently, correspond to **vectors** starting from the origin and pointing to that point (as shown in Figure 1.1).

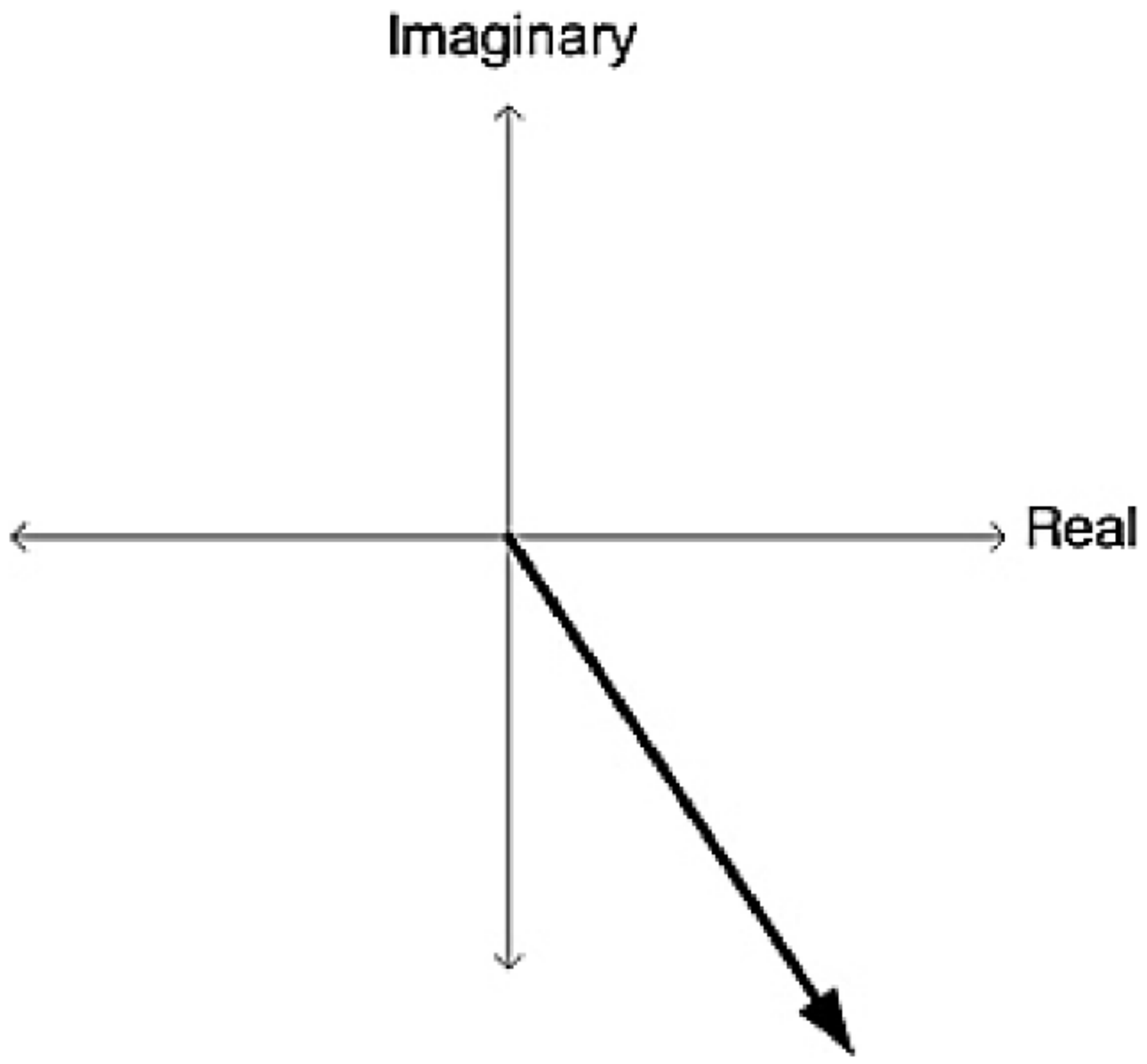



**Figure 1.1.** Complex plane.

In this representation, real numbers (i.e., complex numbers with no imaginary part) sit on the horizontal axis and imaginary numbers sit on the vertical axis. This plane is known as the **complex plane** or the **Argand plane**.

Through this representation, the algebraic properties of the complex numbers can be seen in a new light. Let us start with the modulus: it is nothing more than the **length** of the vector. Indeed, the length of a vector, via Pythagoras' theorem, is the square root of the sum of the squares of its edges, which is precisely the modulus, as defined in the previous section.

**Example 1.3.1** Consider the complex numbers $c = 3 + 4i$ depicted in Figure 1.2. The length of the vector is the hypotenuse of the right triangle whose edges have length 3 and 4, respectively. Pythagoras' theorem gives us the length as

$$length(c) = \sqrt{4^2 + 3^2} = \sqrt{16 + 9} = \sqrt{25} = 5. \tag{1.51}$$

This is exactly the modulus of $c$.

□

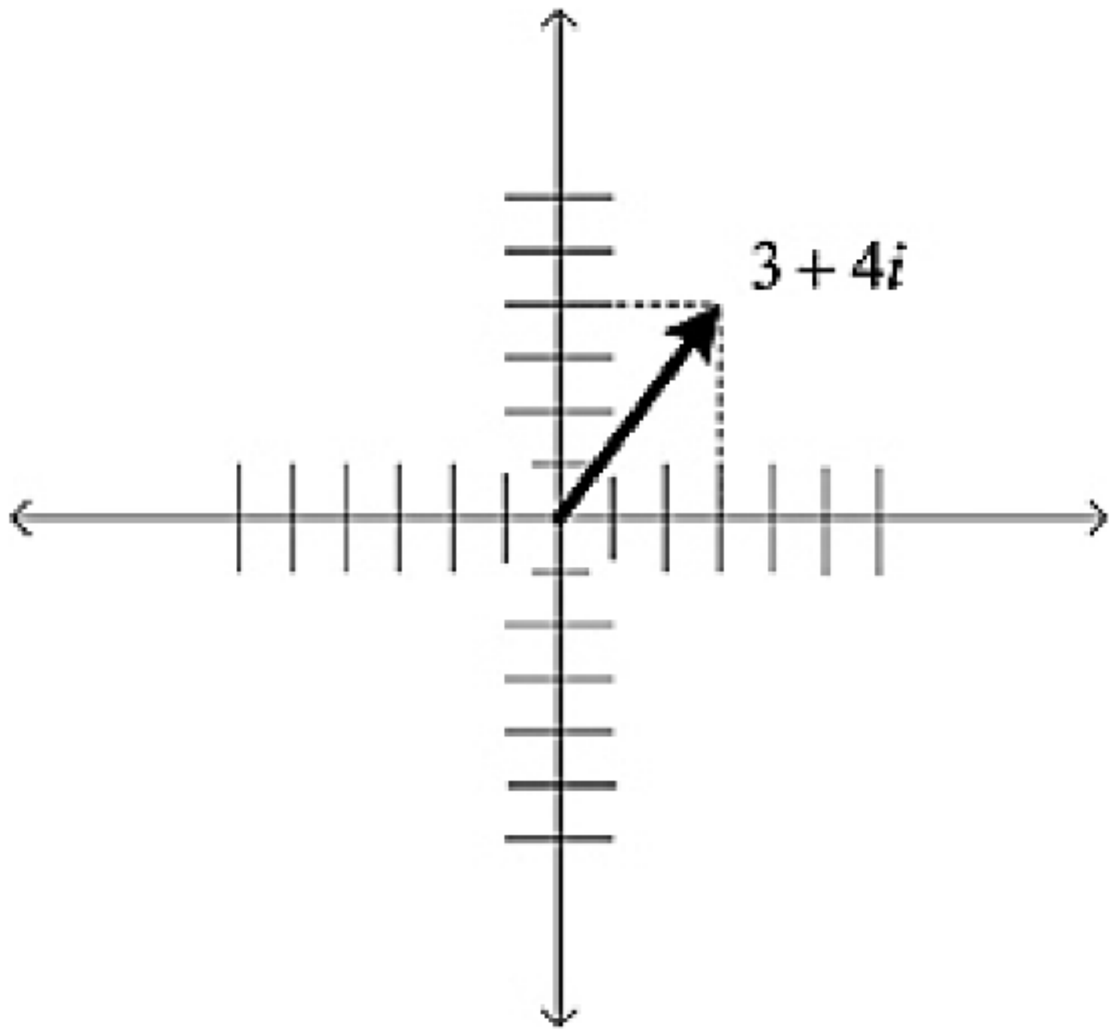



**Figure 1.2.** Vector 3 + 4*i*.

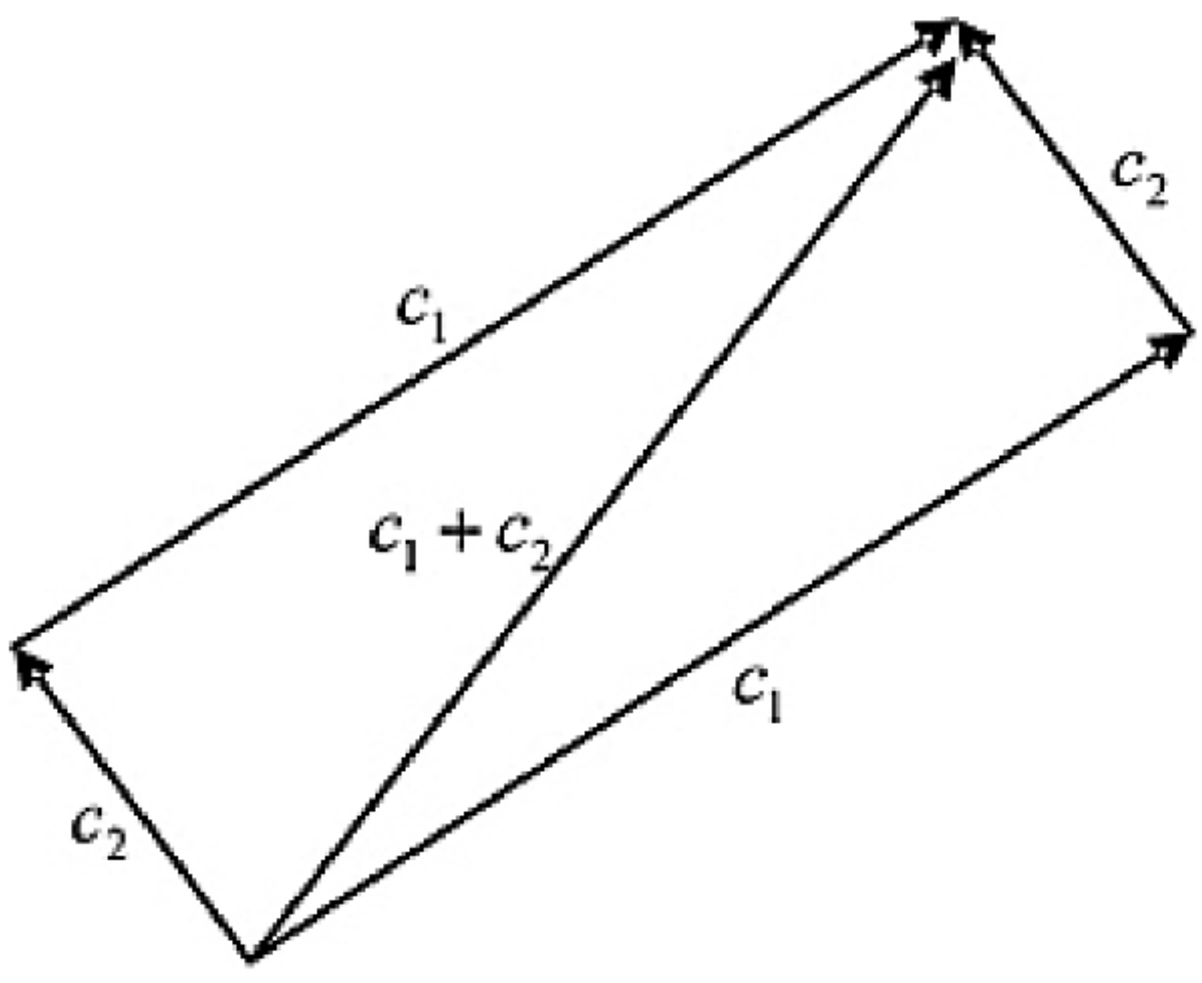



**Figure 1.3.** Parallelogram rule.

Next comes addition: vectors can be added using the so-called **parallelogram rule** illustrated by Figure 1.3. In words, draw the parallelogram whose parallel edges are the two vectors to be added; their sum is the diagonal.

**Exercise 1.3.1** Draw the complex numbers $c_1 = 2 - i$ and $c_2 = 1 + i$ in the complex plane, and add them using the parallelogram rule. Verify that you would get the same result as adding them algebraically (the way we learned in Section 1.2).

■

Subtraction too has a clear geometric meaning: subtracting $c_2$ from $c_1$ is the same as adding the negation of $c_2$, i.e., $-c_2$, to $c_1$. But what is the negation of a vector? It is just the vector of the same length pointed in the opposite direction (see Figure 1.4).

**Exercise 1.3.2** Let $c_1 = 2 - i$ and $c_2 = 1 + i$. Subtract $c_2$ from $c_1$ by first drawing $-c_2$ and then adding it to $c_1$ using the parallelogram rule.

To give a simple geometrical meaning to multiplication, we need to develop yet another characterization of complex numbers. We saw a moment ago that for every complex number we can draw a right triangle, whose edges' lengths are the real and imaginary parts of the number and whose hypotenuse's length is the modulus. Now, suppose someone tells us the modulus of the number what else do we need to know to draw the triangle? The answer is the angle at the origin.

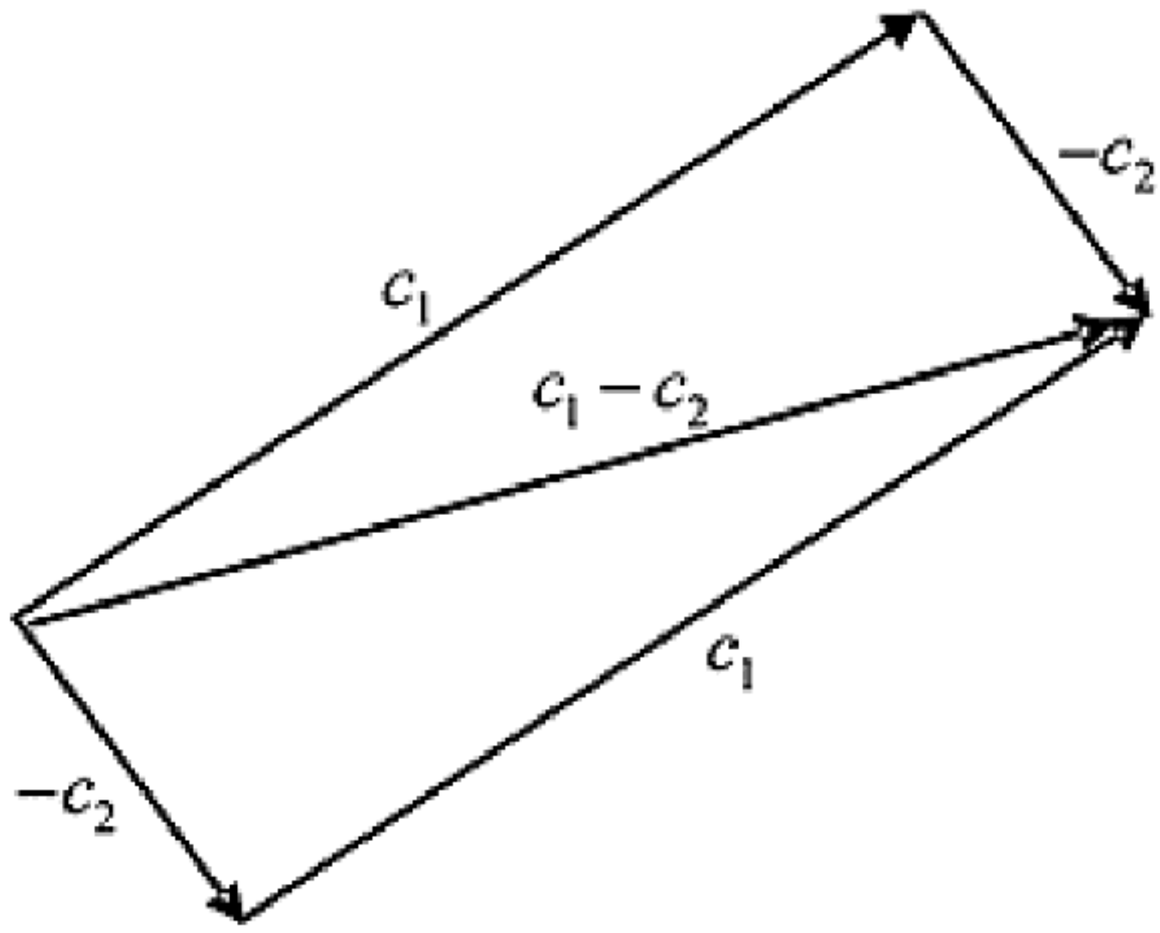



**Figure 1.4.** Subtraction.

The modulus $\rho$ *and* the angle $\theta$ (notice: two real numbers, as before) are enough to uniquely determine the complex number.

$$(a, b) \longmapsto (\rho, \theta). \tag{1.52}$$

We know how to compute $\rho$ from $a$, $b$:

$$\rho = \sqrt{(a^2 + b^2)}. \tag{1.53}$$

$\theta$ is also easy, via trigonometry:

$$\theta = \tan^{-1}\left(\frac{b}{a}\right). \tag{1.54}$$

The ($a$, $b$) representation is known as the **Cartesian representation** of a complex number, whereas ($\rho$, $\theta$) is the **polar representation**.

We can go back from polar to Cartesian representation, again using trigonometry:

$$a = \rho\cos(\theta), \qquad b = \rho\sin(\theta). \tag{1.55}$$

**Example 1.3.2** Let $c = 1 + i$. What is its polar representation?

$$\rho = \sqrt{1^2 + 1^2} = \sqrt{2} \tag{1.56}$$

$$\theta = \tan^{-1}\left(\frac{1}{1}\right) = \tan^{-1}(1) = \frac{\pi}{4} \tag{1.57}$$

$c$ is the vector of length $\sqrt{2}$ from the origin at an angle of $\frac{\pi}{4}$ radians, or 45°.

□

**Exercise 1.3.3** Draw the complex number given by the polar coordinates $\rho = 3$ and $\theta = \frac{\pi}{3}$. Compute its Cartesian coordinates.

■

**Programming Drill 1.3.1** *Write a program that converts a complex number from its Cartesian representation to its polar representation and vice versa.*

Before moving on, let us meditate a little: what kind of insight does the polar representation give us? Instead of providing a ready-made answer, let us begin with a question: how many complex numbers share exactly the same modulus? A moment's thought will tell us that for a *fixed* modulus, say, $\rho = 1$, there is an entire circle centered at the origin (as shown in Figure 1.5).

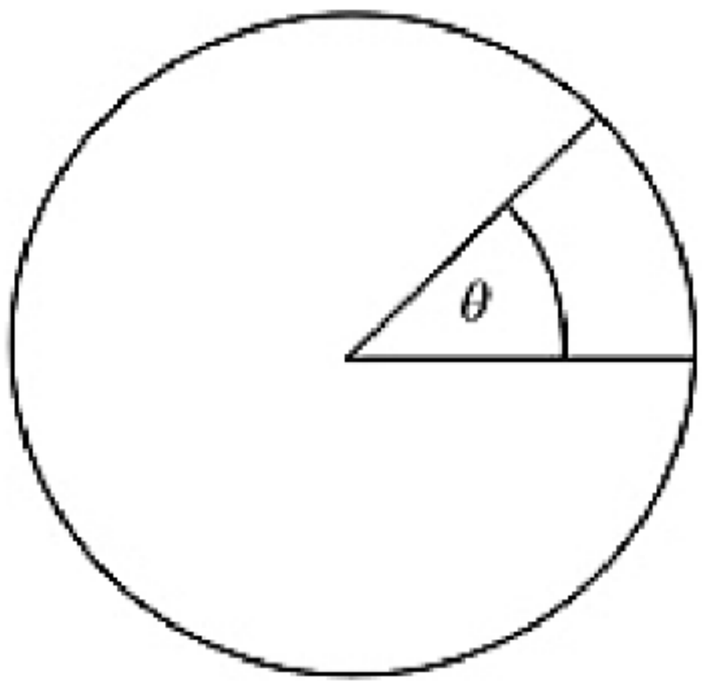



**Figure 1.5.** Phase $\theta$.

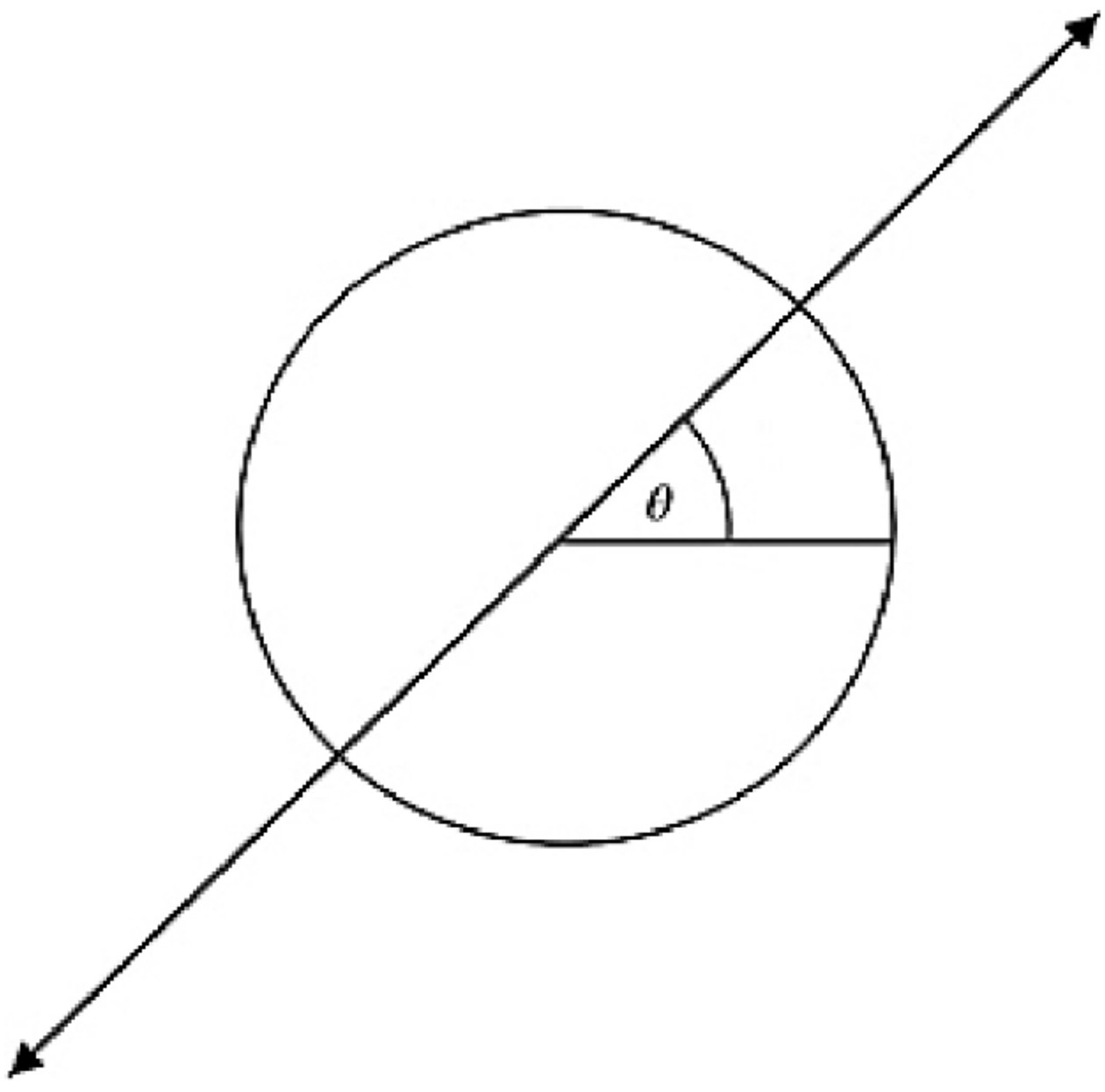



**Figure 1.6.** Points on a line with the same phase.

So, here comes the angle: imagine the circle as your watch, and the complex

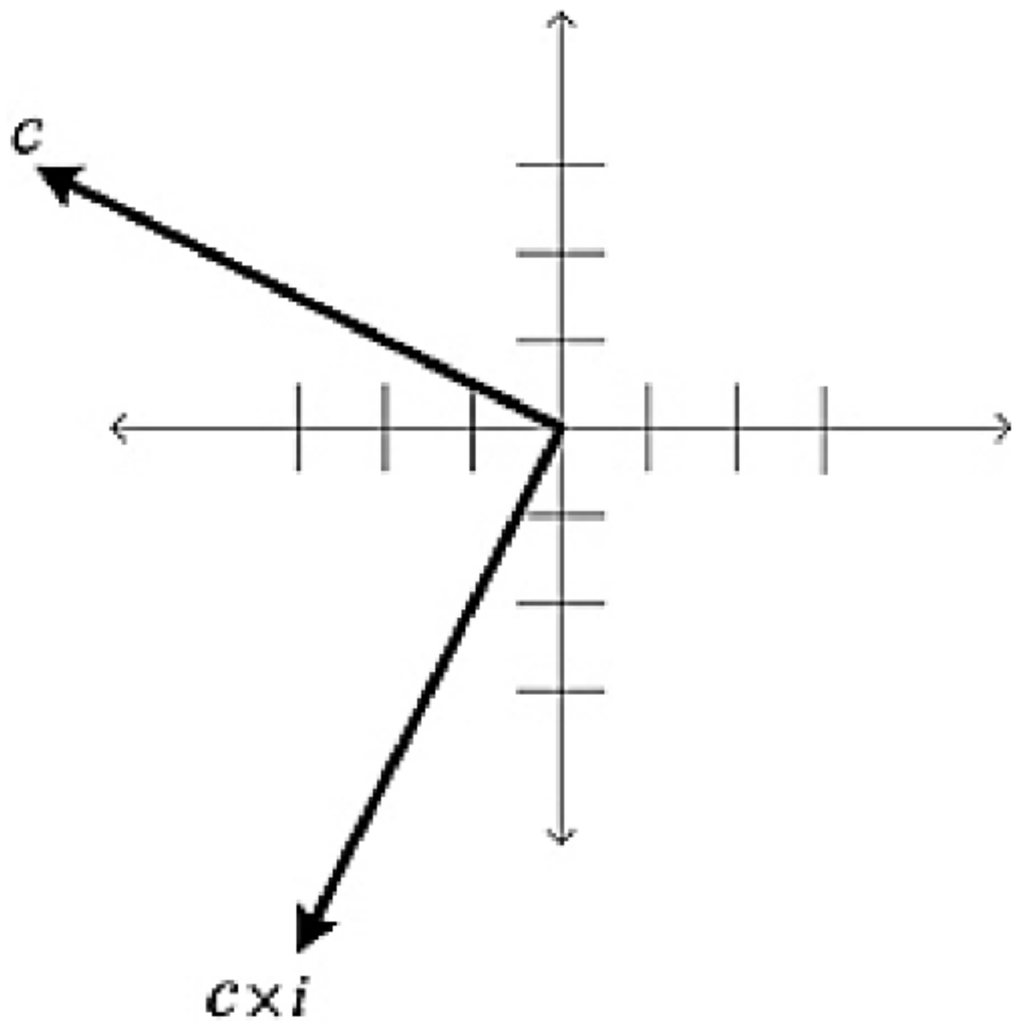



**Figure 1.8.** Multiplication by *i*.

We have implicitly learned an important fact: multiplication in the complex domain has something to do with *rotations* of the complex plane. Indeed, observe just what happens by left or right multiplication by *i*:

$$c \longmapsto c \times i. \tag{1.65}$$

*i* has modulus 1, so the magnitude of the result is exactly equal to that of the starting point. The phase of *i* is $\frac{\pi}{2}$, so multiplying by *i* has the net result of rotating the original complex number by 90°, a right angle. The same happens when we multiply any complex number; so we can safely conclude that multiplication by *i* is a right-angle counterclockwise rotation of the complex plane, as shown in Figure 1.8.

**Exercise 1.3.5** Describe the geometric effect on the plane obtained by multiplying by a real number, i.e., the function

$$c \longmapsto c \times r_0, \tag{1.66}$$

where $r_0$ is a fixed real number.

■

**Exercise 1.3.6** Describe the geometric effect on the plane obtained by multiplying by a generic complex number, i.e., the function

$$c \longmapsto c \times c_0, \tag{1.67}$$

where $c_0$ is a fixed complex number.

■

**Programming Drill 1.3.2** *If you like graphics, write a program that accepts a small drawing around the origin of the complex plane and a complex number. The program should change the drawing by multiplying every point of the diagram by a complex number.*

Now that we are armed with a geometric way of looking at multiplication, we can tackle division as well. After all, division is nothing more than the inverse operation of

multiplication. Assume that

$$c_1 = (\rho_1, \theta_1) \quad \text{and} \quad c_2 = (\rho_2, \theta_2), \tag{1.68}$$

are two complex numbers in polar form; what is the polar form of $\frac{c_1}{c_2}$? A moment's thought tells us that it is the number

$$\frac{c_1}{c_2} = \left(\frac{\rho_1}{\rho_2}, \theta_1 - \theta_2\right). \tag{1.69}$$

In words, we divide the magnitudes and subtract the angles.

**Example 1.3.5** Let $c_1 = -1 + 3i$ and $c_2 = -1 - 4i$. Let us calculate their polar coordinates first:

$$c_1 = \left(\sqrt{(-1)^2 + 3^2}, \tan^{-1}\left(\frac{3}{-1}\right)\right) = (\sqrt{10}, \tan^{-1}(-3)) = (3.1623, 1.8925), \tag{1.70}$$

$$c_2 = \left(\sqrt{(-1)^2 + (-4)^2}, \tan^{-1}\left(\frac{-4}{-1}\right)\right) = (\sqrt{17}, \tan^{-1}(4)) = (4.1231, -1.8158), \tag{1.71}$$

therefore, in polar coordinates the quotient is

$$\frac{c_1}{c_2} = \left(\frac{3.1623}{4.1231}, 1.8925 - (-1.8158)\right) = (0.7670, 3.7083). \tag{1.72}$$

□

**Exercise 1.3.7** Divide $2 + 2i$ by $1 - i$ using both the algebraic and the geometrical method and verify that the results are the same.

■

You may have noticed that in Section 1.2, we have left out two important operations: powers and roots. The reason was that it is much easier to deal with them in the present geometric setting than from the algebraic viewpoint.

Let us begin with powers. If $c = (\rho, \theta)$ is a complex number in polar form and $n$ a positive integer, its $n$th power is just

$$c^n = (\rho^n, n\theta), \tag{1.73}$$

because raising to the $n$th power is multiplying $n$ times. Figure 1.9 shows a complex number and its first, second, and third powers.

**Exercise 1.3.8** Let $c = 1 - i$. Convert it to polar coordinates, calculate its fifth power, and revert the answers to Cartesian coordinates.

■

What happens when the base is a number of magnitude 1? Its powers will also

have magnitude 1; thus, they will stay on the same unit circle. You can think of the various powers 1, 2,… as time units, and a needle moving counterclockwise at constant speed (it covers exactly $\theta$ radians per time unit, where $\theta$ is the phase of the base).

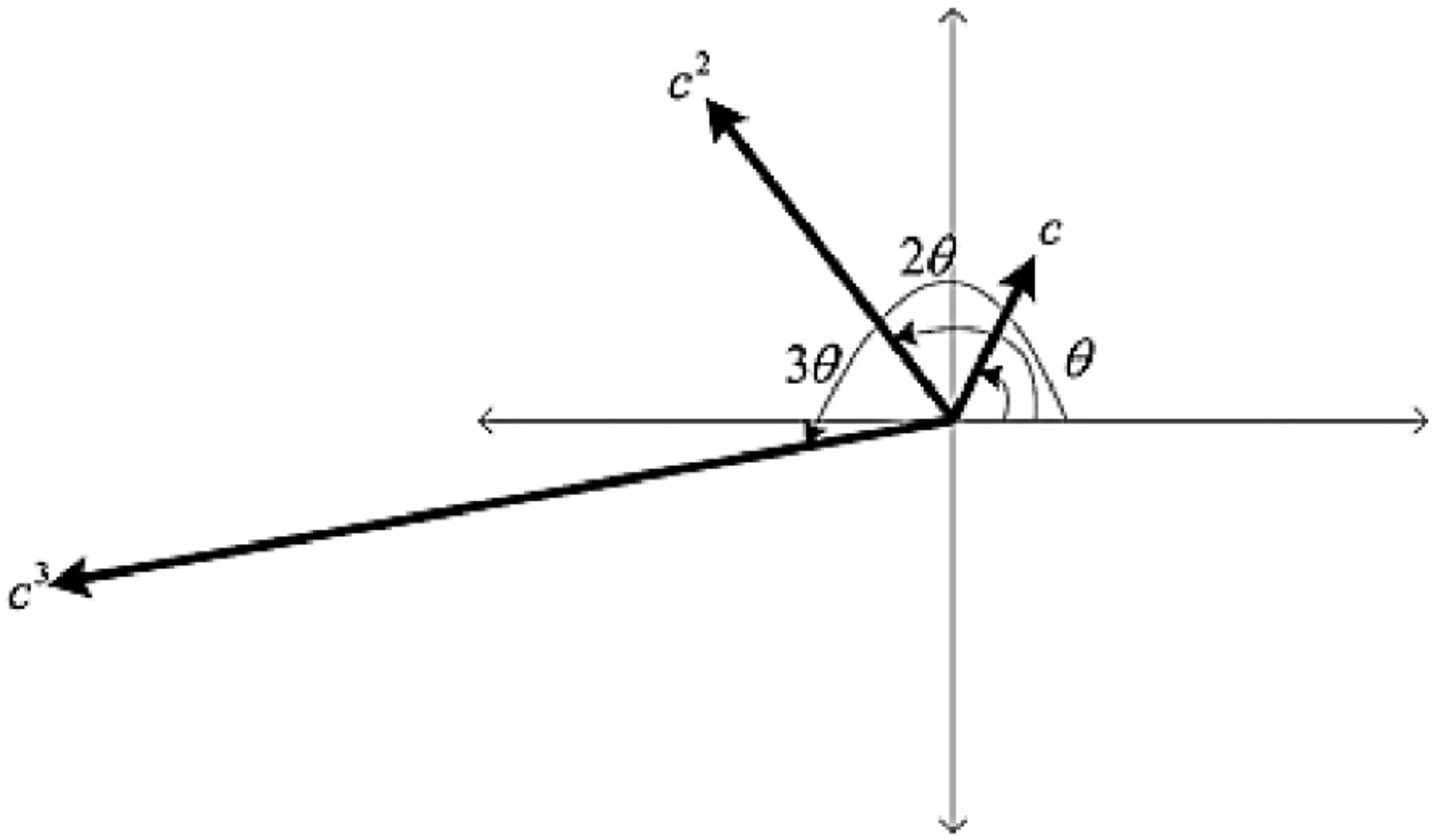



**Figure 1.9.** A complex number and its square and cube.

Let us move on to roots. As you know already from high-school algebra, a root is a fractional power. For instance, the square root means raising the base to the power of one-half; the cube root is raising to the power of one-third; and so forth. The same holds true here, so we may take roots of complex numbers: if $c = (\rho, \theta)$is a complex in polar form, its $n$th root is

$$c^{\frac{1}{n}} = \left(\rho^{\frac{1}{n}}, \frac{1}{n}\theta\right). \tag{1.74}$$

However, things get a bit more complicated. Remember, the phase is defined only up to multiples of $2\pi$. Therefore, we must rewrite Equation (1.74) as

$$c^{\frac{1}{n}} = \left(\sqrt[n]{\rho}, \frac{1}{n}(\theta + k2\pi)\right). \tag{1.75}$$

It appears that there are *several* roots of the same number. This fact should not surprise us: in fact, even among real numbers, roots are not always unique. Take, for instance, the number 2 and notice that there are two square roots, $\sqrt{2}$ and $-\sqrt{2}$.

How many $n$th roots are there? There are precisely $n$th roots for a complex number. Why? Let us go back to Equation (1.75).

$$\frac{1}{n}(\theta + 2k\pi) = \frac{1}{n}\theta + \frac{k}{n}2\pi. \tag{1.76}$$

How many different solutions can we generate by varying $k$? Here they are:

| | |
|---|---|
| $k = 0$ | $\frac{1}{n}\theta$ |
| $k = 1$ | $\frac{1}{n}\theta + \frac{1}{n}2\pi$ |
| $\vdots$ | $\vdots$ |
| $k = n-1$ | $\frac{1}{n}\theta + \frac{n-1}{n}2\pi$ |

(1.77)

That is all: when $k = n$, we obtain the first solution; when $k = n + 1$, we obtain the second solution; and so forth. (Verify this statement!)

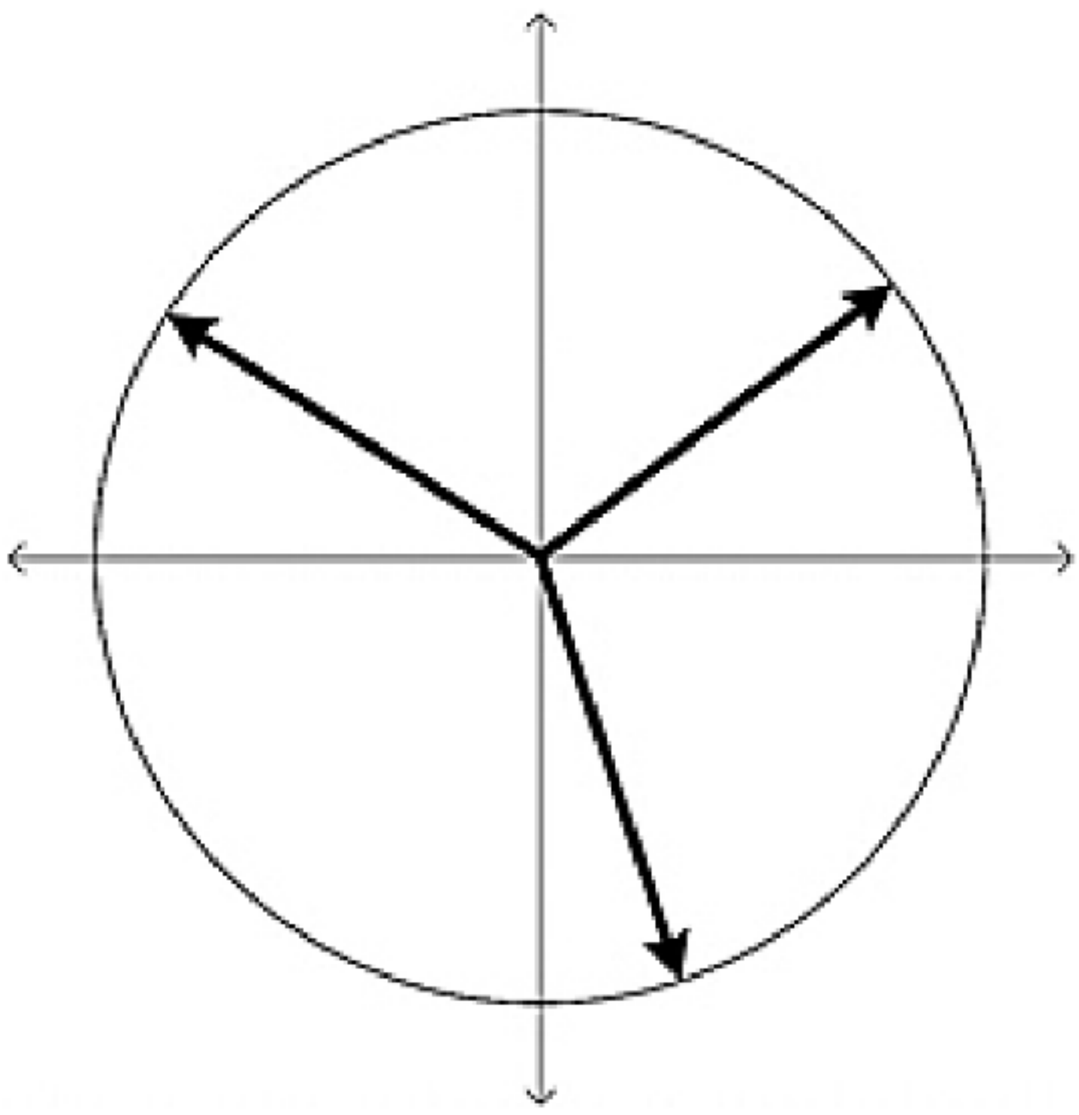


**Figure 1.10.** The three cube roots of unity.

To see what is happening, let us assume that $\rho = 1$; in other words, let us find $n$th roots of a complex number $c = (1, \theta)$ on the unit circle. The $n$ solutions in Equation (1.77) can be interpreted in the following way: Draw the unit circle, and the vectors whose phase is $\frac{1}{n}\theta$, $\frac{1}{n}\theta$ plus an angle equal to $\frac{k}{n}$ of the entire circle, where $k = 1,\ldots, n$. We get precisely the vertices of a regular polygon with $n$ edges. Figure 1.10 is an example when $n = 3$.

**Exercise 1.3.9** Find all the cube roots of $c = 1 + i$.

■

By now we should feel pretty comfortable with the polar representation: we know that any complex number, via the polar-to-Cartesian function, can be written as

$$c = \rho(\cos(\theta) + i\ \sin(\theta)). \tag{1.78}$$

Let us introduce yet another notation that will prove to be very handy in many situations. The starting point is the following formula, known as **Euler's formula**:

$$e^{i\theta} = \cos(\theta) + i\ \sin(\theta). \tag{1.79}$$

The full justification of the remarkable formula of Euler lies outside the scope of this book.[6] However, we can at least provide some evidence that substantiates its validity. First of all, if $\theta = 0$, we get what we expected, namely, 1. Secondly,

$$\begin{aligned} e^{i(\theta_1+\theta_2)} &= \cos(\theta_1+\theta_2) + i\sin(\theta_1+\theta_2) \\ &= \cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2) \\ &\quad + i((\sin(\theta_1)\cos(\theta_2) + \sin(\theta_2)\cos(\theta_1)) \\ &= (\cos(\theta_1) + i\sin(\theta_1))((\cos(\theta_2) + i\sin(\theta_2)) \\ &= e^{i\theta_1} \times e^{i\theta_2}. \end{aligned} \tag{1.83}$$

In other words, the exponential function takes sums into products as it does in the real case.

**Exercise 1.3.10** Prove **De Moivre's formula**:

$$(e^{\theta i})^n = \cos(n\theta) + i\sin(n\theta). \tag{1.84}$$

(Hint: The trigonometric identities used earlier, with induction on $n$, will do the work.) ■

Now that we know how to take the exponential of an imaginary number, there is no problem in defining the exponential of an arbitrary complex number:

$$e^{a+bi} = e^a \times e^{bi} = e^a(\cos(b) + i\sin(b)). \tag{1.85}$$

Euler's formula enables us to rewrite Equation (1.78) in a more compact form:

$$c = \rho\, e^{i\theta}. \tag{1.86}$$

We shall refer to Equation (1.86) as the **exponential form** of a complex number.

**Exercise 1.3.11** Write the number $c = 3 - 4i$ in exponential form.

■

The exponential notation simplifies matters when we perform multiplication:

$$c_1c_2 = \rho_1\, e^{i\theta_1}\rho_2\, e^{i\theta_2} = \rho_1\rho_2\, e^{i(\theta_1+\theta_2)}. \tag{1.87}$$

**Exercise 1.3.12** Rewrite the law for dividing complex numbers in exponential form.

■

With this notation, we can look at the roots of the complex number $1 = (1, 0) = 1 + 0i$. Let $n$ be a fixed number. There are $n$ different **roots of unity**. Setting $c = (1, 0)$. in Equation (1.75), we get

$$c^{\frac{1}{n}} = (1,0)^{\frac{1}{n}} = \left(\sqrt[n]{1}, \frac{1}{n}(0 + 2k\pi) = \left(1, \frac{2k\pi}{n}\right)\right). \tag{1.88}$$

■

**Exercise 1.3.19** Show that each Möbius transformation has an inverse that is also a Möbius transformation, i.e., for each $R_{a,b,c,d}$ you can find $R_{a',b',c',d'}$ such that

$$R_{a',b',c',d'} \circ R_{a,b,c,d}(x) = x. \tag{1.100}$$

■

There are many more functions in the complex domain, but to introduce them one needs tools from **complex analysis**, i.e., calculus over the complex numbers. The main idea is quite simple: replace polynomials with a power series, i.e., polynomials with an infinite number of terms. The functions one studies are the so-called **analytic functions**, which are functions that can be coherently pieced together from small parts, each of which is represented by a series.

**Programming Drill 1.3.3** *Expand your program. Add functions for multiplication, division, and returning the polar coordinates of a number.*

We have covered the basic language of complex numbers. Before we embark on our quantum journey, we need another tool: vector spaces over the complex field.

---

**References:** Most of the material found in this chapter can be found in any calculus or linear algebra textbook. References for some of the more advanced material presented at the end of the chapter can be found in, e.g., Bak and Newman (1996), Needham (1999), Schwerdtfeger (1980), and Silverman (1984).

The history of complex numbers goes back to the mid-sixteenth century during the Italian Renaissance. The story of Tartaglia, Cardano, Bombelli and their effort to solve algebraic equations is well worth reading. Some of this fascinating tale is in Nahin (1998), Mazur (2002), and several wonderful sections in Penrose (1994).

[1] For the German-speaking reader, here is the original text (the translation at the beginning is ours):
Du, hast du das vorhin ganz verstanden?
Was?
Die Geschichte mit den imaginären Zahlen?
Musil's *Törless* is a remarkable book. A substantial part is dedicated to the struggle of young Törless to come to grips with mathematics, as well as with his own life. Definitely recommended!

[2] The definition given in Equation (1.40) is entirely equivalent to the more familiar one: $|a| = a$ if $a \geq 0$, and $|a| = -a$ if $a < 0$.

[3] A subset of a field that is a field in its own right is called a **subfield**: $\mathbb{R}$ is a subfield of $\mathbb{C}$.

[4] Its "geometric" name is **real-axis reflection**. The name becomes obvious in the next section.

[5] In the geometric viewpoint, it is known as **imaginary-axis reflection**. After reading Section 1.3, we invite you to investigate this operation a bit further.

[6] For the calculus-savvy reader: Use the well-known Taylor expansions.

$$e^x = 1 + x + \frac{x^2}{2} + \cdots + \frac{x^n}{n!} + \cdots, \tag{1.80}$$

$$\sin(x) = x - \frac{x^3}{3!} + \cdots + \frac{(-1)^n}{(2n+1)!}x^{2n+1} + \cdots, \tag{1.81}$$

$$\cos(x) = 1 - \frac{x^2}{2} + \cdots + \frac{(-1)^n}{(2n)!}x^{2n} + \cdots. \tag{1.82}$$

Assume that they hold for complex values of $x$. Now, formally multiply $\sin(x)$ by $i$ and add componentwise $\cos(x)$ to obtain Euler's formula.

[7] Möbius transformations are a truly fascinating topic, and perhaps the best entrance door to the geometry of complex numbers. We invite you to find out more about them in Schwerdtfeger (1980).

# 2
# Complex Vector Spaces

*Philosophy is written in that great book which continually lies open before us (I mean the Universe). But one cannot understand this book until one has learned to understand the language and to know the letters in which it is written. It is written in the language of mathematics, and the letters are triangles, circles and other geometric figures. Without these means it is impossible for mankind to understand a single word; without these means there is only vain stumbling in a dark labyrinth.*[1]

Galileo Galilei

Quantum theory is cast in the language of complex vector spaces. These are mathematical structures that are based on complex numbers. We learned all that we need about such numbers in Chapter 1. Armed with this knowledge, we can now tackle complex vector spaces themselves.

Section 2.1 goes through the main example of a (finite-dimensional) complex vector space at tutorial pace. Section 2.2 provides formal definitions, basic properties, and more examples. Each of Section 2.3 through Section 2.7 discusses an advanced topic.

---

**Reader Tip.** The reader might find some of this chapter to be "just boring math." If you are eager to leap into the quantum world, we suggest reading the first two or three sections before moving on to Chapter 3. Return to Chapter 2 as a reference when needed (using the index and the table of contents to find specific topics). ♡

---

A small disclaimer is in order. The theory of complex vector spaces is a vast and beautiful subject. Lengthy textbooks have been written on this important area of mathematics. It is impossible to provide anything more than a small glimpse into the beauty and profundity of this topic in one chapter. Rather than "teaching" our reader complex vector spaces, we aim to cover the bare minimum of concepts, terminology, and notation needed in order to start quantum computing. It is our sincere hope that reading this chapter will inspire further investigation into this remarkable subject.

## 2.1 $\mathbb{C}^n$ AS THE PRIMARY EXAMPLE

The primary example of a complex vector space is the set of vectors (one-dimensional arrays) of a fixed length with complex entries. These vectors will describe the states of quantum systems and quantum computers. In order to fix our ideas and to see clearly what type of structure this set has, let us carefully examine one concrete example: the set of vectors of length 4. We shall denote this set as $\mathbb{C}^4 = \mathbb{C} \times \mathbb{C} \times \mathbb{C} \times \mathbb{C}$, which reminds us that each vector is an ordered list of four complex numbers.

A typical element of $\mathbb{C}^4$ looks like this:

$$\begin{bmatrix} 6-4i \\ 7+3i \\ 4.2-8.1i \\ -3i \end{bmatrix}. \tag{2.1}$$

We might call this vector $V$. We denote the $j$th element of $V$ as $V[j]$. The top row is row number 0 (not 1);[2] hence, $V[1] = 7 + 3i$.

What types of operations can we carry out with such vectors? One operation that seems obvious is to form the **addition** of two vectors. For example, given two vectors of $\mathbb{C}^4$

$$V = \begin{bmatrix} 6-4i \\ 7+3i \\ 4.2-8.1i \\ -3i \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} 16+2.3i \\ -7i \\ 6 \\ -4i \end{bmatrix}, \tag{2.2}$$

we can add them to form $V + W \in \mathbb{C}^4$ by adding their respective entries:

$$\begin{bmatrix} 6-4i \\ 7+3i \\ 4.2-8.1i \\ -3i \end{bmatrix} + \begin{bmatrix} 16+2.3i \\ -7i \\ 6 \\ -4i \end{bmatrix} = \begin{bmatrix} (6-4i)+(16+2.3i) \\ (7+3i)+(-7i) \\ (4.2-8.1i)+(6) \\ (-3i)+(-4i) \end{bmatrix} = \begin{bmatrix} 22-1.7i \\ 7-4i \\ 10.2-8.1i \\ -7i \end{bmatrix}. \tag{2.3}$$

Formally, this operation amounts to

$$(V+W)[j] = V[j] + W[j]. \tag{2.4}$$

**Exercise 2.1.1** Add the following two vectors:

$$(-W)[j] = -(W[j]). \tag{2.13}$$

The set $\mathbb{C}^4$ with the addition, inverse operations, and zero such that the addition is associative and commutative, form something called an **Abelian group**.

What other structure does our set $\mathbb{C}^4$ have? Take an arbitrary complex number, say, $c = 3 + 2i$. Call this number a **scalar**. Take a vector

$$V = \begin{bmatrix} 6+3i \\ 0+0i \\ 5+1i \\ 4 \end{bmatrix}. \tag{2.14}$$

We can **multiply an element by a scalar** by multiplying the scalar with each entry of the vector; i.e.,

$$(3+2i) \cdot \begin{bmatrix} 6+3i \\ 0+0i \\ 5+1i \\ 4 \end{bmatrix} = \begin{bmatrix} 12+21i \\ 0+0i \\ 13+13i \\ 12+8i \end{bmatrix}. \tag{2.15}$$

Formally, for a complex number $c$ and a vector $V$, we form $c \cdot V$, which is defined as

$$(c \cdot V)[j] = c \times V[j], \tag{2.16}$$

where the $\times$ is complex multiplication. We shall omit the $\cdot$ when the scalar multiplication is understood.

**Exercise 2.1.3** Scalar multiply $8-2i$ with $\begin{bmatrix} 16+2.3i \\ -7i \\ 6 \\ 5-4i \end{bmatrix}$.

■

Scalar multiplication satisfies the following properties: for all $c, c_1, c_2 \in \mathbb{C}$ and for all $V, W \in \mathbb{C}^4$,

- $1 \cdot V = V$,
- $c_1 \cdot (c_2 \cdot V) = (c_1 \times c_2) \cdot V$,
- $c \cdot (V + W) = c \cdot V + c \cdot W$,
- $(c_1 + c_2) \cdot V = c_1 \cdot V + c_2 \cdot V$.

**Exercise 2.1.4** Formally prove that $(c_1 + c_2) \cdot V = c_1 \cdot V + c_2 \cdot V$.

■

An Abelian group with a scalar multiplication that satisfies these properties is called a **complex vector space**.

Notice that we have been working with vectors of size 4. However, everything that we have stated about vectors of size 4 is also true for vectors of arbitrary size. So the set $\mathbb{C}^n$ for a fixed but arbitrary $n$ also has the structure of a complex vector space. In fact, these vector spaces will be the primary examples we will be working with for the rest of the book.

**Programming Drill 2.1.1** *Write three functions that perform the addition, inverse, and scalar multiplication operations for $\mathbb{C}^n$, i.e., write a function that accepts the appropriate input for each of the operations and outputs the vector.*

## 2.2 DEFINITIONS, PROPERTIES, AND EXAMPLES

There are many other examples of complex vector spaces. We shall need to broaden our horizon and present a formal definition of a complex vector space.

**Definition 2.2.1** *A* **complex vector space** *is a nonempty set $\mathbb{V}$, whose elements we shall call vectors, with three operations*

- *Addition:* $+ : \mathbb{V} \times \mathbb{V} \longrightarrow \mathbb{V}$
- *Negation:* $- : \mathbb{V} \longrightarrow \mathbb{V}$
- *Scalar multiplication:* $\cdot : \mathbb{C} \times \mathbb{V} \longrightarrow \mathbb{V}$

*and a distinguished element called the* **zero vector** $\mathbf{0} \in \mathbb{V}$ *in the set. These operations and zero must satisfy the following properties: for all V, W,* $X \in \mathbb{V}$ *and for all c,* $c_1, c_2 \in \mathbb{C}$,

(i) *Commutativity of addition:* $V + W = W + V$,
(ii) *Associativity of addition:* $(V + W) + X = V + (W + X)$,
(iii) *Zero is an additive identity:* $V + \mathbf{0} = V = \mathbf{0} + V$,
(iv) *Every vector has an inverse:* $V + (-V) = \mathbf{0} = (-V) + V$,
(v) *Scalar multiplication has a unit:* $1 \cdot V = V$,
(vi) *Scalar multiplication respects complex multiplication:*

$$c_1 \cdot (c_2 \cdot V) = (c_1 \times c_2) \cdot V, \tag{2.17}$$

(vii) *Scalar multiplication distributes over addition:*

$$c \cdot (V + W) = c \cdot V + c \cdot W, \tag{2.18}$$

(viii) *Scalar multiplication distributes over complex addition:*

$$(c_1 + c_2) \cdot V = c_1 \cdot V + c_2 \cdot V. \tag{2.19}$$

To recap, any set that has an addition operation, an inverse operation, and a zero element that satisfies Properties (i), (ii), (iii), and (iv) is called an **Abelian group**. If, furthermore, there is a scalar multiplication operation that satisfies all the properties, then the set with the operations is called a **complex vector space**.

Although our main concern is complex vector spaces, we can gain much intuition from real vector spaces.

**Definition 2.2.2** *A* **real vector space** *is a nonempty set $\mathbb{V}$ (whose elements we shall call vectors), along with an addition operation and a negation operation. Most important, there is a scalar multiplication that uses $\mathbb{R}$ and not*

$\mathbb{C}$, *i.e.*,

$$\cdot : \mathbb{R} \times \mathbb{V} \longrightarrow \mathbb{V}. \tag{2.20}$$

*This set and these operations must satisfy the analogous properties of a complex vector space.*

In plain words, a real vector space is like a complex vector space except that we only require the scalar multiplication to be defined for scalars in $\mathbb{R} \subset \mathbb{C}$. From the fact that $\mathbb{R} \subset \mathbb{C}$, it is easy to see that for every $\mathbb{V}$ we have $\mathbb{R} \times \mathbb{V} \subset \mathbb{C} \times \mathbb{V}$. If we have a given

$$\cdot : \mathbb{C} \times \mathbb{V} \longrightarrow \mathbb{V}, \tag{2.21}$$

then we can write

$$\mathbb{R} \times \mathbb{V} \hookrightarrow \mathbb{C} \times \mathbb{V} \longrightarrow \mathbb{V}. \tag{2.22}$$

We conclude that every complex vector space can automatically be given a real vector space structure.

Let us descend from the abstract highlands and look at some concrete examples.

**Example 2.2.1** $\mathbb{C}^n$, the set of vectors of length $n$ with complex entries, is a complex vector space that serves as our primary example for the rest of the book. In Section 2.1, we exhibited the operations and described the properties that are satisfied.

□

**Example 2.2.2** $\mathbb{C}^n$, the set of vectors of length $n$ with complex entries, is also a real vector space because every complex vector space is also a real vector space. The operations are the same as those in Example 2.2.1.

□

**Example 2.2.3** $\mathbb{R}^n$, the set of vectors of length $n$ with real number entries, is a real vector space. Notice that there is no obvious way to make this into a complex vector space. What would the scalar multiplication of a complex number with a real vector be?

□

In Chapter 1, we discussed the geometry of $\mathbb{C} = \mathbb{C}^1$. We showed how every complex number can be thought of as a point in a two-dimensional plane. Things get more complicated for $\mathbb{C}^2$. Every element of $\mathbb{C}^2$ involves two complex numbers or four real numbers. One could visualize this as an element of four-dimensional space. However, the human brain is not equipped to visualize four-dimensional space. The most we can deal with is three dimensions. Many times throughout this text, we shall discuss $\mathbb{C}^n$ and then revert to $\mathbb{R}^3$ in order to develop an intuition for what is going on.

It pays to pause for a moment to take an in-depth look at the geometry of $\mathbb{R}^3$. Every vector of $\mathbb{R}^3$ can be thought of as a point in three-dimensional space or equivalently, as an arrow from the origin of $\mathbb{R}^3$ to that point. So the vector $\begin{bmatrix} 5 \\ -7 \\ 6.3 \end{bmatrix}$ shown in Figure 2.1 is 5 units in the $x$ direction, $-7$ units in the $y$ direction, and 6.3 units in the $z$ direction.

Given two vectors $V = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \end{bmatrix}$ and $V' = \begin{bmatrix} r'_0 \\ r'_1 \\ r'_2 \end{bmatrix}$ of $\mathbb{R}^3$, we may add them to form $\begin{bmatrix} r_0 + r'_0 \\ r_1 + r'_1 \\ r_2 + r'_2 \end{bmatrix}$. Addition can be seen as making a parallelogram in $\mathbb{R}^3$ where you attach the beginning of one arrow to the end of the other one. The result of the addition is the composition of the arrows (see Figure 2.2). The reason that we can be ambiguous about which arrow comes first demonstrates the commutativity property of addition.

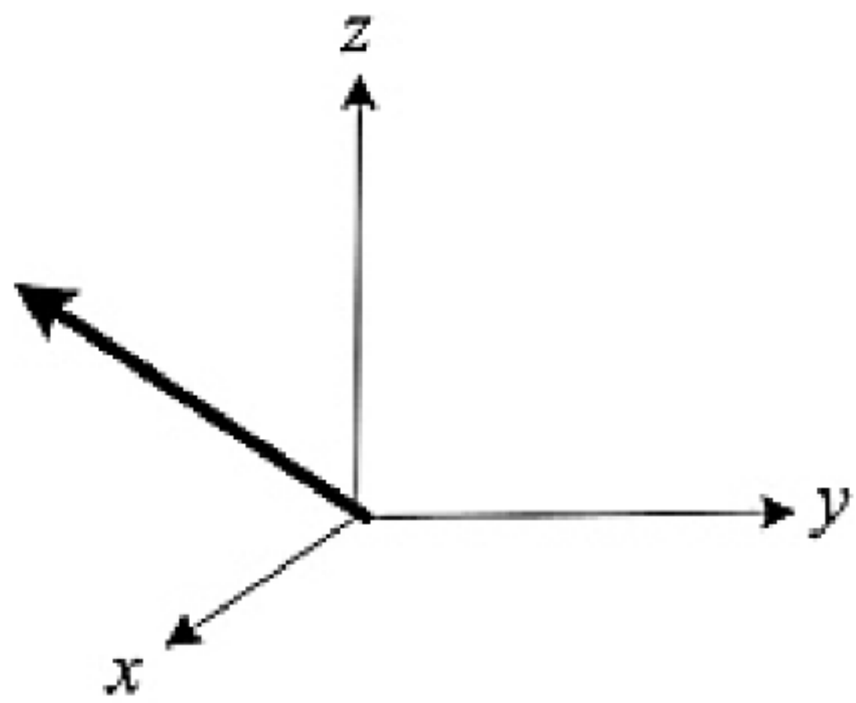



**Figure 2.1.** A vector in three dimensional space.

Given a vector $V = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \end{bmatrix}$ in $\mathbb{R}^3$, we form the inverse $-V = \begin{bmatrix} -r_0 \\ -r_1 \\ -r_2 \end{bmatrix}$ by looking at the arrow in the opposite direction with respect to all dimensions (as in Figure 2.3).

And finally, the scalar multiplication of a real number $r$ and a vector $V = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \end{bmatrix}$ is $r$. $r \cdot V = \begin{bmatrix} rr_0 \\ rr_1 \\ rr_2 \end{bmatrix}$, which is simply the vector $V$ stretched or shrunk by $r$ (as in Figure 2.4).

It is useful to look at some of the properties of a vector space from the geometric point of view. For example, consider the property $r \cdot (V + W) = r \cdot V + r \cdot W$. This corresponds to Figure 2.5.

**Exercise 2.2.1** Let $r_1 = 2$, $r_2 = 3$, and $V = \begin{bmatrix} 2 \\ -4 \\ 1 \end{bmatrix}$. Verify Property (vi), i.e., calculate $r_1 \cdot (r_2 \cdot V)$ and $(r_1 \times r_2) \cdot V$ and show that they coincide.

■

**Exercise 2.2.2** Draw pictures in $\mathbb{R}^3$ that explain Properties (vi) and (viii) of the definition of a real vector space.

■

Let us continue our list of examples.

**Example 2.2.4** $\mathbb{C}^{m\times n}$, the set of all *m*-by-*n* matrices (two-dimensional arrays) with complex entries, is a complex vector space.

□

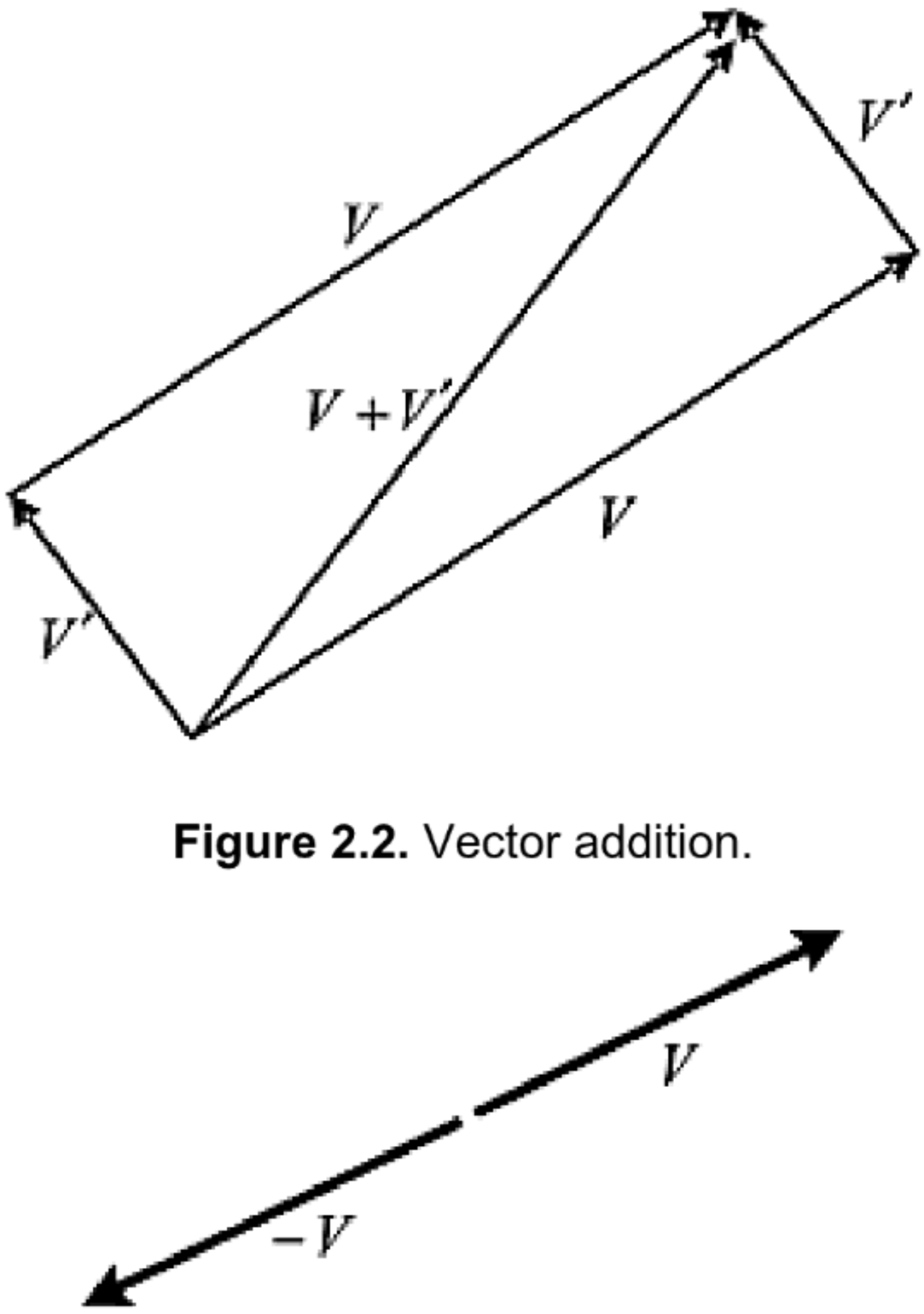



**Figure 2.2.** Vector addition.

**Figure 2.3.** Inverse of a vector.

For a given $A \in \mathbb{C}^{m\times n}$, we denote the complex entry in the *j*th row and the *k*th column as $A[j, k]$ or $c_{j,k}$. We shall denote the *j*th row as $A[j, -]$ and the *k*th column as $A[-, k]$. Several times throughout the text we shall show the row and column numbers explicitly to the left and top of the square brackets:

$$A = \begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \\ \vdots \\ \boldsymbol{m-1} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{0} & \mathbf{1} & \cdots & \boldsymbol{n-1} \end{array} \\ \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m-1,0} & c_{m-1,1} & \cdots & c_{m-1,n-1} \end{bmatrix} \end{array}. \tag{2.23}$$

The operations for $\mathbb{C}^{m\times n}$ are given as follows: Addition is

$$c \cdot \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m-1,0} & c_{m-1,1} & \cdots & c_{m-1,n-1} \end{bmatrix}$$

$$= \begin{bmatrix} c \times c_{0,0} & c \times c_{0,1} & \cdots & c \times c_{0,n-1} \\ c \times c_{1,0} & c \times c_{1,1} & \cdots & c \times c_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ c \times c_{m-1,0} & c \times c_{m-1,1} & \cdots & c \times c_{m-1,n-1} \end{bmatrix}. \tag{2.26}$$

Formally, these operations can be described by the following formulas:
For two matrices, $A$, $B \in \mathbb{C}^{m \times n}$, we add them as

$$(A + B)[j, k] = A[j, k] + B[j, k]. \tag{2.27}$$

The inverse of $A$ is

$$(-A)[j, k] = -(A[j, k]). \tag{2.28}$$

The scalar multiplication of $A$ with a complex number $c \in \mathbb{C}$ is

$$(c \cdot A)[j, k] = c \times A[j, k]. \tag{2.29}$$

**Exercise 2.2.3** Let $c_1 = 2i$, $c_2 = 1 + 2i$, and $A = \begin{bmatrix} 1-i & 3 \\ 2+2i & 4+i \end{bmatrix}$. Verify Properties (vi) and (viii) in showing $\mathbb{C}^{2\times 2}$ is a complex vector space.

■

**Exercise 2.2.4** Show that these operations on $\mathbb{C}^{m \times n}$ satisfy Properties (v), (vi), and (viii) of being a complex vector space.

■

**Programming Drill 2.2.1** *Convert your functions from the last programming drill so that instead of accepting elements of $\mathbb{C}^n$, they accept elements of $\mathbb{C}^{m \times n}$.*

When $n = 1$, the matrices $\mathbb{C}^{m \times n} = \mathbb{C}^{m \times 1} = \mathbb{C}^m$, which we dealt with in Section 2.1. Thus, we can think of vectors as special types of matrices.

When $m = n$, the vector space $\mathbb{C}^{n \times n}$ has more operations and more structure than just a complex vector space. Here are three operations that one can perform on an $A \in \mathbb{C}^{n \times n}$:

- The **transpose** of $A$, denoted $A^T$, is defined as

$$A^T[j,k] = A[k,j]. \tag{2.30}$$

- The **conjugate** of $A$, denoted $\overline{A}$, is the matrix in which each element is the complex conjugate of the corresponding element of the original matrix,[3] i.e., $\overline{A}[j,k] = \overline{A[j,k]}$.
- The transpose operation and the conjugate operation are combined to form the **adjoint** or **dagger** operation. The adjoint of $A$, denoted as $A\dagger$, is defined as $A^\dagger = (\overline{A})^T = \overline{(A^T)}$ or $A^\dagger[j,k] = \overline{A[k,j]}$.

**Exercise 2.2.5** Find the transpose, conjugate, and adjoint of

$$\begin{bmatrix} 6-3i & 2+12i & -19i \\ 0 & 5+2.1i & 17 \\ 1 & 2+5i & 3-4.5i \end{bmatrix}. \tag{2.31}$$

■

These three operations are defined even when $m \neq n$. The transpose and adjoint are both functions from $\mathbb{C}^{m\times n}$ to $\mathbb{C}^{n\times m}$.

These operations satisfy the following properties for all $c \in \mathbb{C}$ and for all $A$, $B \in \mathbb{C}^{m\times n}$:

(i) Transpose is idempotent: $(A^T)^T = A$.
(ii) Transpose respects addition: $(A + B)^T = A^T + B^T$.
(iii) Transpose respects scalar multiplication: $(c \cdot A)^T = c \cdot A^T$.
(iv) Conjugate is idempotent: $\overline{\overline{A}} = A$.
(v) Conjugate respects addition: $\overline{A + B} = \overline{A} + \overline{B}$.
(vi) Conjugate respects scalar multiplication: $\overline{c \cdot A} = \overline{c} \cdot \overline{A}$.
(vii) Adjoint is idempotent: $(A\dagger)\dagger = A$.
(viii) Adjoint respects addition: $(A + B)\dagger = A\dagger + B\dagger$.
(ix) Adjoint relates to scalar multiplication: $(c \cdot A)^\dagger = \overline{c} \cdot A^\dagger$.

**Exercise 2.2.6** Prove that conjugation respects scalar multiplication, i.e., $\overline{c \cdot A} = \overline{c} \cdot \overline{A}$.

■

**Exercise 2.2.7** Prove Properties (vii), (viii), and (ix) using Properties (i) – (vi).

■

The transpose shall be used often in the text to save space. Rather than writing

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} \tag{2.32}$$

which requires more space, we write $[c_0, c_1, \ldots, c_{n-1}]^T$.

When $m = n$, there is another binary operation that is used: **matrix multiplication**. Consider the following two 3-by-3 matrices:

$$A = \begin{bmatrix} 3+2i & 0 & 5-6i \\ 1 & 4+2i & i \\ 4-i & 0 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 2-i & 6-4i \\ 0 & 4+5i & 2 \\ 7-4i & 2+7i & 0 \end{bmatrix}. \tag{2.33}$$

We form the matrix product of $A$ and $B$, denoted $A * B$. $A * B$ will also be a 3-by-3 matrix. $(A * B)[0, 0]$ will be found by multiplying each element of the 0th row of $A$ with the corresponding element of the 0th column of $B$. We then sum the results:

$$\begin{aligned} (A \star B)[0,0] &= ((3+2i) \times 5) + (0 \times 0) + ((5-6i) \times (7-4i)) \\ &= (15+10i) + (0) + (11-62i) = 26-52i. \end{aligned} \tag{2.34}$$

The $(A * B)[j, k]$ entry can be found by multiplying each element of $A[j, -]$ with the appropriate element of $B[-, k]$ and summing the results. So,

$$(A \star B) = \begin{bmatrix} 26-52i & 60+24i & 26 \\ 9+7i & 1+29i & 14 \\ 48-21i & 15+22i & 20-22i \end{bmatrix}. \tag{2.35}$$

**Exercise 2.2.8** Find $B * A$. Does it equal $A * B$?

■

Matrix multiplication is defined in a more general setting. The matrices do not have to be square. Rather, the number of columns in the first matrix must be the same as the number of rows in the second one. Matrix multiplication is a binary operation

$$\star : \mathbb{C}^{m \times n} \times \mathbb{C}^{n \times p} \longrightarrow \mathbb{C}^{m \times p}. \tag{2.36}$$

Formally, given $A$ in $\mathbb{C}^{m \times n}$ and $B$ in $\mathbb{C}^{n \times p}$, we construct $A * B$ in $\mathbb{C}^{m \times p}$ as

$$(A \star B)[j,k] = \sum_{h=0}^{n-1} (A[j,h] \times B[h,k]). \tag{2.37}$$

When the multiplication is understood, we shall omit the *.

For every $n$, there is a special $n$-by-$n$ matrix called the **identity matrix**,

$$I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{2.38}$$

that plays the role of a unit of matrix multiplication. When $n$ is understood, we shall omit it.

Matrix multiplication satisfies the following properties: For all $A$, $B$, and $C$ in $\mathbb{C}^{n\times n}$,

(i) Matrix multiplication is associative: $(A * B) * C = A * (B * C)$.
(ii) Matrix multiplication has $I_n$ as a unit: $I_n * A = A = A * I_n$.
(iii) Matrix multiplication distributes over addition:

$$A \star (B + C) = (A \star B) + (A \star C), \tag{2.39}$$

$$(B + C) \star A = (B \star A) + (C \star A). \tag{2.40}$$

(iv) Matrix multiplication respects scalar multiplication:

$$c \cdot (A \star B) = (c \cdot A) \star B = A \star (c \cdot B). \tag{2.41}$$

(v) Matrix multiplication relates to the transpose:

$$(A \star B)^T = B^T \star A^T. \tag{2.42}$$

(vi) Matrix multiplication respects the conjugate:

$$\overline{A \star B} = \overline{A} \star \overline{B}. \tag{2.43}$$

(vii) Matrix multiplication relates to the adjoint:

$$(A \star B)^\dagger = B^\dagger \star A^\dagger. \tag{2.44}$$

Notice that commutativity is *not* a basic property of matrix multiplication. This fact will be very important in quantum mechanics.

**Exercise 2.2.9** Prove Property (v) in the above list.

■

**Exercise 2.2.10** Use $A$ and $B$ from Equation (2.33) and show that $(A * B)^\dagger = B\dagger$

* $A\dagger$.

■

**Exercise 2.2.11** Prove Property (vii) from Properties (v) and (vi).

■

**Definition 2.2.3** *A complex vector space* $\mathbb{V}$ *with a multiplication * that satisfies the first four properties is called a* **complex algebra**.

**Programming Drill 2.2.2** *Write a function that accepts two complex matrices of the appropriate size. The function should do matrix multiplication and return the result.*

Let $A$ be any element in $\mathbb{C}^{n\times n}$. Then for any element $B \in \mathbb{C}^n$, we have that $A * B$ is in $\mathbb{C}^n$. In other words, multiplication by $A$ gives one a function from $\mathbb{C}^n$ to $\mathbb{C}^n$. From Equations (2.39) and (2.41), we see that this function preserves addition and scalar multiplication. We will write this map as $A : \mathbb{C}^n \longrightarrow \mathbb{C}^n$.

Let us look ahead for a moment and see what relevance this abstract mathematics has for quantum computing. Just as $\mathbb{C}^n$ has a major role, the complex algebra $\mathbb{C}^{n\times n}$ shall also be in our cast of characters. The elements of $\mathbb{C}^n$ are the ways of describing the states of a quantum system. Some suitable elements of $\mathbb{C}^{n\times n}$ will correspond to the changes that occur to the states of a quantum system. Given a state $X \in \mathbb{C}^n$ and a matrix $A \in \mathbb{C}^{n\times n}$, we shall form another state of the system $A * X$ which is an element of $\mathbb{C}^n$[4]. Formally, * in this case is a function *: $\mathbb{C}^{n\times n} \times \mathbb{C}^n \longrightarrow \mathbb{C}^n$. We say that the algebra of matrices "**acts**" on the vectors to yield new vectors. We shall see this **action** again and again in the following chapters.

**Programming Drill 2.2.3** *Write a function that accepts a vector and a matrix and outputs the vector resulting from the*"*action*."

We return to our list of examples.

**Example 2.2.5** $\mathbb{C}^{m\times n}$, the set of all *m*-by-*n* matrices (two-dimensional arrays) with complex entries, is a real vector space. (Remember: Every complex vector space is also a real vector space.)

□

**Example 2.2.6** $\mathbb{R}^{m\times n}$, the set of all *m*-by-*n* matrices (two-dimensional arrays) with real entries, is a real vector space.

□

**Definition 2.2.4** *Given two complex vector spaces* $\mathbb{V}$ *and* $\mathbb{V}'$*, we say that* $\mathbb{V}$ *is a* **complex subspace** *of* $\mathbb{V}'$ *if* $\mathbb{V}$ *is a subset of* $\mathbb{V}'$ *and the operations of* $\mathbb{V}$ *are restrictions of operations of* $\mathbb{V}'$.

Equivalently, $\mathbb{V}$ is a complex subspace of $\mathbb{V}'$ if $\mathbb{V}$ is a subset of the set $\mathbb{V}'$ and

(i) $\mathbb{V}$ is closed under addition: For all $V_1$ and $V_2$ in $\mathbb{V}$, $V_1 + V_2 \in \mathbb{V}$.
(ii) $\mathbb{V}$ is closed under scalar multiplication: For all $c \in \mathbb{C}$ and $V \in \mathbb{V}$, $c \cdot V \in \mathbb{V}$.

$$(f+g)(n) = f(n)+g(n). \tag{2.53}$$

The additive inverse of $f$ is

$$(-f)(n) = -(f(n)). \tag{2.54}$$

The scalar multiple of $c \in \mathbb{C}$ and $f$ is the function

$$(c \cdot f)(n) = c \times f(n). \tag{2.55}$$

Because the operations are determined by their values at each of their "points" in the input, the constructed functions are said to be constructed **pointwise**.

□

**Exercise 2.2.15** Show that $Func(\mathbb{N}, \mathbb{C})$ with these operations forms a complex vector space.

■

**Example 2.2.12** We can generalize $Func(\mathbb{N}, \mathbb{C})$ to other sets of functions. For any $a < b$ in $\mathbb{R}$, the set of functions from the interval $[a, b] \subseteq \mathbb{R}$ to $\mathbb{C}$ denoted $Func([a, b], \mathbb{R})$ is a complex vector space.

□

**Exercise 2.2.16** Show that $Func(\mathbb{N}, \mathbb{R})$ and $Func([a, b], \mathbb{R})$ are real vector spaces.

■

**Example 2.2.13** There are several ways of constructing new vector spaces from existing ones. Here we see one method and Section 2.7 describes another. Let $(\mathbb{V}, +, -, \mathbf{0}, \cdot)$ and $(\mathbb{V}', +', -', \mathbf{0}', \cdot')$ be two complex vector spaces. We construct a new complex vector space $(\mathbb{V} \times \mathbb{V}', +'', -'', \mathbf{0}'', \cdot'')$ called the **Cartesian product**[5] or the **direct sum** of $\mathbb{V}$ and $\mathbb{V}'$. The vectors are ordered pairs of vectors $(V, V') \in \mathbb{V} \times \mathbb{V}'$. Operations are performed pointwise:

$$(V_1, V_1') +'' (V_2, V_2') = (V_1 + V_2, V_1' +' V_2'), \tag{2.56}$$

$$-''(V, V') = (-V, -'V'), \tag{2.57}$$

$$\mathbf{0}'' = (\mathbf{0}, \mathbf{0}'), \tag{2.58}$$

$$c \cdot'' (V, V') = (c \cdot V, c \cdot' V'). \tag{2.59}$$

□

**Exercise 2.2.17** Show that $\mathbb{C}^m \times \mathbb{C}^n$ is isomorphic to $\mathbb{C}^{m+n}$.

■

**Exercise 2.2.18** Show that $\mathbb{C}^m$ and $\mathbb{C}^n$ are each a complex subspace of $\mathbb{C}^m \times \mathbb{C}^n$.

■

## *2.3 BASIS AND DIMENSION*

A basis of a vector space is a set of vectors of that vector space that is special in the sense that all other vectors can be uniquely written in terms of these basis vectors.

**Definition 2.3.1** *Let $\mathbb{V}$ be a complex (real) vector space. $V \in \mathbb{V}$ is a* **linear combination** *of the vectors $V_0, V_1, \ldots, V_{n-1}$ in $\mathbb{V}$ if V can be written as*

$$V = c_0 \cdot V_0 + c_1 \cdot V_1 + \cdots + c_{n-1} \cdot V_{n-1} \tag{2.60}$$

*for some $c_0, c_1, \ldots, c_{n-1}$ in $\mathbb{C}$ ($\mathbb{R}$).*

Let us return to $\mathbb{R}^3$ for examples.

**Example 2.3.1** As

$$3\begin{bmatrix}5\\-2\\3\end{bmatrix} + 5\begin{bmatrix}0\\1\\4\end{bmatrix} - 4\begin{bmatrix}-6\\1\\0\end{bmatrix} + 2.1\begin{bmatrix}3\\1\\1\end{bmatrix} = \begin{bmatrix}45.3\\-2.9\\31.1\end{bmatrix}, \tag{2.61}$$

we say that

$$[45.3, -2.9, 31.1]^T \tag{2.62}$$

is a linear combination of

$$\begin{bmatrix}5\\-2\\3\end{bmatrix}, \begin{bmatrix}0\\1\\4\end{bmatrix}, \begin{bmatrix}-6\\1\\0\end{bmatrix}, \quad \text{and} \quad \begin{bmatrix}3\\1\\1\end{bmatrix}. \tag{2.63}$$

□

**Definition 2.3.2** *A set $\{V_0, V_1, \ldots, V_{n-1}\}$ of vectors in $\mathbb{V}$ is called* **linearly independent** *if*

$$\mathbf{0} = c_0 \cdot V_0 + c_1 \cdot V_1 + \cdots + c_{n-1} \cdot V_{n-1} \tag{2.64}$$

*implies that $c_0 = c_1 = \ldots = c_{n-1} = 0$. This means that the only way that a linear combination of the vectors can be the zero vector is if all the $c_j$ are zero.*

It can be shown that this definition is equivalent to saying that for any nonzero $V \in \mathbb{V}$, there are *unique* coefficients $c_0, c_1, \ldots, c_{n-1}$ in $\mathbb{C}$ such that

$$V = c_0 \cdot V_0 + c_1 \cdot V_1 + \cdots + c_{n-1} \cdot V_{n-1}. \tag{2.65}$$

The set of vectors are called linearly independent because each of the vectors in the set $\{V_0, V_1, \ldots, V_{n-1}\}$ cannot be written as a combination of the others in the set.

**Example 2.3.2** The set of vectors

$$\left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\} \tag{2.66}$$

is linearly independent because the only way that

$$\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = x \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.67}$$

can occur is if $0 = x$, $0 = x + y$, and $0 = x + y + z$. By substitution, we see that $x = y = z = 0$.

□

**Example 2.3.3** The set of vectors

$$\left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} \right\} \tag{2.68}$$

is not linearly independent (called **linearly dependent**) because

$$\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = x \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + z \begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} \tag{2.69}$$

can happen when $x = 2$, $y = -3$, and $z = -1$.

□

**Exercise 2.3.1** Show that the set of vectors

$$\left\{\begin{bmatrix}1\\2\\3\end{bmatrix}, \begin{bmatrix}3\\0\\2\end{bmatrix}, \begin{bmatrix}1\\-4\\-4\end{bmatrix}\right\} \tag{2.70}$$

is not linearly independent.

■

**Definition 2.3.3** *A set* $\mathcal{B} = \{V_0, V_1, \ldots, V_{n-1}\} \subseteq \mathbb{V}$ *of vectors is called a* **basis** *of a* (*complex*) *vector space* $\mathbb{V}$ *if both*

(i) *every,* $V \in \mathbb{V}$ *can be written as a linear combination of vectors from* $\mathcal{B}$ *and*
(ii) $\mathcal{B}$ *is linearly independent.*

**Example 2.3.4** $\mathbb{R}^3$ has a basis

$$\left\{\begin{bmatrix}1\\1\\1\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right\}. \tag{2.71}$$

□

**Exercise 2.3.2** Verify that the preceding three vectors are in fact a basis of $\mathbb{R}^3$ .

■

There may be many sets that each form a basis of a particular vector space but there is also a basis that is easier to work with called the **canonical basis** or the **standard basis**. Many of the examples that we will deal with have canonical basis. Let us look at some examples of canonical basis.

■ $\mathbb{R}^3$:

$$\left\{\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right\}. \tag{2.72}$$

■ $\mathbb{C}^n$ (and $\mathbb{R}^n$):

$$M_{\mathcal{D}\leftarrow\mathcal{B}} = \begin{bmatrix} 2 & -\frac{3}{2} \\ -3 & \frac{5}{2} \end{bmatrix}. \tag{2.90}$$

So

$$V_{\mathcal{D}} = M_{\mathcal{D}\leftarrow\mathcal{B}} V_{\mathcal{B}} = \begin{bmatrix} 2 & -\frac{3}{2} \\ -3 & \frac{5}{2} \end{bmatrix} \begin{bmatrix} 3 \\ -2 \end{bmatrix} = \begin{bmatrix} 9 \\ -14 \end{bmatrix}. \tag{2.91}$$

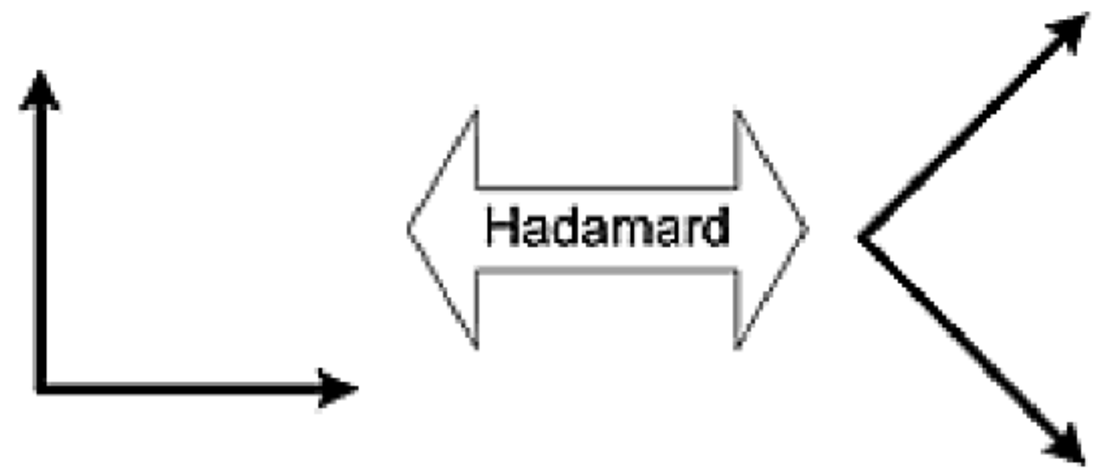



**Figure 2.6.** The Hadamard matrix as a transition between two bases.

Checking, we see that

$$\begin{bmatrix} 7 \\ -17 \end{bmatrix} = 9 \begin{bmatrix} -7 \\ 9 \end{bmatrix} - 14 \begin{bmatrix} -5 \\ 7 \end{bmatrix}. \tag{2.92}$$

□

Given two bases of a finite-dimensional vector space, there are standard algorithms to find a transition matrix from one to the other. (We will not need to know how to find these matrices.)

In $\mathbb{R}^2$, the transition matrix from the canonical basis

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \tag{2.93}$$

to this other basis

$$\left\{ \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \right\} \tag{2.94}$$

is the **Hadamard matrix**: