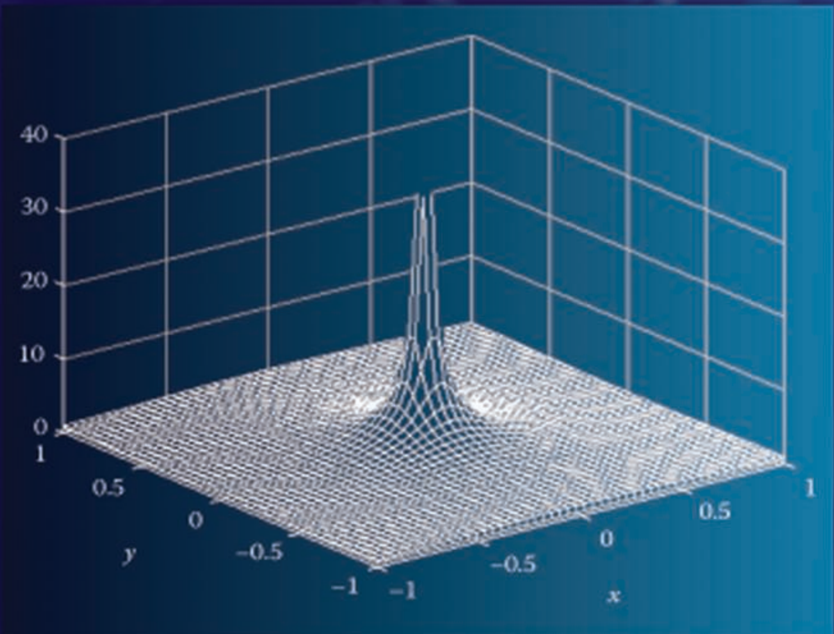
TEXTBOOKS IN MATHEMATICS

A MATLAB® COMPANION TO COMPLEX VARIABLES



A. David Wunsch

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

A MATLAB® COMPANION TO COMPLEX VARIABLES

TEXTBOOKS in MATHEMATICS

Series Editors: Al Boggess and Ken Rosen

PUBLISHED TITLES

ABSTRACT ALGEBRA: AN INTERACTIVE APPROACH, SECOND EDITION
William Paulsen

ABSTRACT ALGEBRA: AN INQUIRY-BASED APPROACH
Jonathan K. Hodge, Steven Schlicker, and Ted Sundstrom

ADVANCED LINEAR ALGEBRA
Hugo Woerdeman

APPLIED ABSTRACT ALGEBRA WITH MAPLE™ AND MATLAB®, THIRD EDITION
Richard Klima, Neil Sigmon, and Ernest Stitzinger

APPLIED DIFFERENTIAL EQUATIONS: THE PRIMARY COURSE
Vladimir Dobrushkin

COMPUTATIONAL MATHEMATICS: MODELS, METHODS, AND ANALYSIS WITH MATLAB® AND MPI, SECOND EDITION
Robert E. White

DIFFERENTIAL EQUATIONS: THEORY, TECHNIQUE, AND PRACTICE, SECOND EDITION
Steven G. Krantz

DIFFERENTIAL EQUATIONS: THEORY, TECHNIQUE, AND PRACTICE WITH BOUNDARY VALUE PROBLEMS
Steven G. Krantz

DIFFERENTIAL EQUATIONS WITH MATLAB®: EXPLORATION, APPLICATIONS, AND THEORY
Mark A. McKibben and Micah D. Webster

ELEMENTARY NUMBER THEORY
James S. Kraft and Lawrence C. Washington

EXPLORING LINEAR ALGEBRA: LABS AND PROJECTS WITH MATHEMATICA®
Crista Arangala

GRAPHS & DIGRAPHS, SIXTH EDITION
Gary Chartrand, Linda Lesniak, and Ping Zhang

INTRODUCTION TO ABSTRACT ALGEBRA, SECOND EDITION
Jonathan D. H. Smith

PUBLISHED TITLES CONTINUED

INTRODUCTION TO MATHEMATICAL PROOFS: A TRANSITION TO ADVANCED MATHEMATICS, SECOND EDITION
Charles E. Roberts, Jr.

INTRODUCTION TO NUMBER THEORY, SECOND EDITION
Marty Erickson, Anthony Vazzana, and David Garth

LINEAR ALGEBRA, GEOMETRY AND TRANSFORMATION
Bruce Solomon

MATHEMATICAL MODELLING WITH CASE STUDIES: USING MAPLE™ AND MATLAB®, THIRD EDITION
B. Barnes and G. R. Fulford

MATHEMATICS IN GAMES, SPORTS, AND GAMBLING–THE GAMES PEOPLE PLAY, SECOND EDITION
Ronald J. Gould

THE MATHEMATICS OF GAMES: AN INTRODUCTION TO PROBABILITY
David G. Taylor

MEASURE THEORY AND FINE PROPERTIES OF FUNCTIONS, REVISED EDITION
Lawrence C. Evans and Ronald F. Gariepy

NUMERICAL ANALYSIS FOR ENGINEERS: METHODS AND APPLICATIONS, SECOND EDITION
Bilal Ayyub and Richard H. McCuen

ORDINARY DIFFERENTIAL EQUATIONS: AN INTRODUCTION TO THE FUNDAMENTALS
Kenneth B. Howell

RISK ANALYSIS IN ENGINEERING AND ECONOMICS, SECOND EDITION
Bilal M. Ayyub

TRANSFORMATIONAL PLANE GEOMETRY
Ronald N. Umble and Zhigang Han

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

TEXTBOOKS in MATHEMATICS

A MATLAB® COMPANION TO COMPLEX VARIABLES

A. David Wunsch
University of Massachusetts Lowell

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20160315

International Standard Book Number-13: 978-1-4987-5567-2 (Paperback)

Library of Congress Cataloging-in-Publication Data

Names: Wunsch, A. David.
Title: A Matlab companion to complex variables / A. David Wunsch.
Description: Boca Raton : Taylor & Francis, 2016. | Series: Textbooks in mathematics ; 41 | "A CRC title." | Includes bibliographical references and index.
Identifiers: LCCN 2015044810 | ISBN 9781498755672 (alk. paper)
Subjects: LCSH: Functions of complex variables--Data processing. | MATLAB.
Classification: LCC QA331.7 .W865 2016 | DDC 515/.9028553--dc23
LC record available at http://lccn.loc.gov/2015044810

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Acknowledgments xi
Introduction xiii
A Note to the Reader xv

1. Complex Arithmetic 1
1.1 The Rectangular Form 1
1.1.1 A Caveat on Complex Numbers as Matrix Elements 3
Exercises 4
1.2 The Polar Form of Complex Numbers 5
Exercises 8
1.3 Fractional Powers of Complex Numbers 9
1.3.1 Using **sqrt** 11
1.3.2 Reminder: A Warning about Square Roots and Fractional Powers in MATLAB 11
1.3.3 Use of **roots** 11
1.3.4 A Program to Give You Fractional Roots and to Do a Check 14
1.3.5 A Further Caveat with Fractional Powers: The **Plot** Function 14
Exercises 19
1.4 Complex Symbolic Algebra 21
1.4.1 Numbers as Symbols 22
Exercises 24

2. Loci and Regions in the Complex Plane and Displaying Complex Functions 27
2.1 **Meshgrid** and Three-Dimensional Plotting 27
Exercises 33
2.2 Two-Dimensional Plots: The Contour Plot 34
Exercises 38
2.3 Displaying Regions in the Complex Plane 39
2.3.1 A Note on **pcolor** 43
Exercises 44
2.4 Three-Dimensional and Contour Plots for Functions with Singularities 45
2.4.1 Plots When There Are Isolated Singular Points 46
2.4.1.1 Removable Singular Points 46
2.4.1.2 Pole Singularities 48
2.4.1.3 Essential Singularities 52
2.4.1.4 Branch Cut and Branch Point Singularities: Three-Dimensional Plots 54

Exercises ... 58
2.5 Contour Plots Affected by Branch Cuts ... 60
Exercises ... 66

3. Sequences, Series, Limits, and Integrals ... 69
3.1 Sequences ... 69
Exercises ... 76
3.2 Infinite Series and Their Convergence ... 77
3.2.1 Using MATLAB to Obtain the Series Coefficients ... 82
3.2.2 Laurent Series ... 84
Exercises ... 90
3.3 Integration in the Complex Plane: Part 1, Finite Sums as Approximations ... 93
Exercises ... 98
3.4 Integrations in the Complex Plane: Part 2, **int** and **quadgk** ... 100
3.4.1 Integration in the Complex Plane with MATLAB ... 104
3.4.2 **Waypoints** ... 105
Exercises ... 107

4. Harmonic Functions, Conformal Mapping, and Some Applications ... 113
4.1 Introduction ... 113
Exercises ... 125
4.2 The Bilinear Transformation ... 127
4.2.1 The Cross-Ratio ... 131
Exercises ... 135
4.3 Laplace's Equation, Harmonic Functions: Voltage, Temperature, and Fluid Flow ... 138
Exercises ... 150
4.4 Complex Potentials, Cauchy–Riemann Equations, the Stream Function, and Streamlines ... 152
4.4.1 Cauchy–Riemann Equations ... 152
Exercises ... 160
4.5 Mapping, Dirichlet, and Neumann Problems and Line Sources ... 162
4.5.1 Dirichlet Problems ... 162
4.5.2 A Dirichlet Problem ... 169
4.5.3 A Neumann Problem ... 173
4.5.4 Line Sources and Conformal Mapping ... 176
Exercises ... 182

5. Polynomials, Roots, the Principle of the Argument, and Nyquist Stability ... 195
5.1 Introduction ... 195
5.2 The Fundamental Theorem of Algebra ... 195
5.2.1 MATLAB **roots** ... 196

5.2.2 Use of **Zeros** ... 200
5.2.3 **Roots** versus **Factor** ... 201
5.2.4 Derivatives of Polynomials and the Function Roots ... 202
Exercises ... 205
5.3 The Principle of the Argument ... 208
5.3.1 Rouché's Theorem ... 219
Exercises ... 221
5.4 Nyquist Plots, the Location of Roots, and the Pole-Zero Map ... 223
5.4.1 The Use of **pzmap** ... 230
5.4.2 A Caveat on the Use of **pzmap** ... 231
Exercises ... 233

6. Transforms: Laplace, Fourier, Z, and Hilbert ... 235
6.1 Introduction ... 235
6.2 The Laplace Transform ... 235
6.2.1 Branch Points and Branch Cuts in the Laplace Transform ... 239
6.2.2 Heaviside and Dirac Delta Functions ... 240
6.2.3 The Inverse Laplace Transform ... 246
Exercises ... 251
6.3 The Fourier Transform ... 257
6.3.1 Changing Variables: Fourier to Laplace and Vice Versa ... 266
6.3.2 Convolution of Functions ... 267
Exercises ... 269
6.4 The Z Transform ... 271
6.4.1 The Z Transform of a Product ... 290
6.4.2 Inverse Z Transform of a Product and Convolutions ... 293
Exercises ... 295
6.5 The Hilbert Transform ... 299
6.5.1 Definition ... 300
Exercises ... 306

7. Coda: Fractals and the Mandelbrot Set ... 309
7.1 Julia Sets ... 323
Appendices to Coda ... 326
Appendix A ... 326
Appendix B ... 328
Appendix C ... 329
Exercises ... 330

Index ... 335

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Acknowledgments

In writing this book, I consulted with a number of very generous people. One is my brother, Carl Wunsch, formerly of MIT and now at Harvard. He is a physical oceanographer and knows his fluid dynamics and helped me with mine. His son, Jared, who is in the mathematics faculty at Northwestern University, answered some questions about fractals without indicating that I should have been able to figure these out for myself. He is a good mathematician and a tactful teacher.

Kenneth Falconer, who is in the mathematics department at the University of St. Andrews, has written a very graceful introduction to the subject of fractals published by Oxford University Press. He has kindly answered some of my e-mail inquiries on this subject. And equally generous has been Professor Bodil Branner of Denmark, the first woman to head the Danish Mathematical Society, who replied to my e-mails. I can recommend her essay, The Mandelbrot Set, which appeared in the book *Chaos and Fractals* published by the American Mathematical Society.

Michael F. Brown ably assisted me on the last two editions of my book on complex variables and kindly found time to help me with this text. He is a good mathematician and a careful reader.

Finally, MathWorks, publisher of MATLAB®, provides a team of consultants to help authors who are writing MATLAB-based books. These people have responded quickly to my e-mails and have almost always managed to correct my work and supply helpful advice.

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098, USA
Tel: 508-647-7000
Fax: 508-647-7001
E-mail: info@mathworks.com
Web: www.mathworks.com

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Introduction

I hope the reader will enjoy using this book. It is for someone with at least a beginner's knowledge of MATLAB® who is learning the branch of advanced calculus called "functions of a complex variable." Complex variable theory is the calculus of functions dependent on variables that can assume complex numerical values. I want you to discover that MATLAB is your friend when you are learning this type of mathematics.

One of the daunting aspects of complex numbers (and variables) is that simple arithmetic operations that are easily done in one's head, if they involve real quantities, become tedious when the numbers are complex. You can ask a class of third graders for the product of the numbers 3, 4, and 5, and someone with the answer will raise his or her hand in less than 5 seconds. Ask a college student for the product of the numbers $(1 + 3i)$, $(1 + 4i)$, $(1 + 5i)$ and he or she will, after a little squirming, probably resort to pencil and paper, although the problem is solvable in one's head with some serious concentration. If you were to ask me this question as I'm typing this sentence, I would open the MATLAB window on my computer and type in the product. Here we see a glimpse of the utility of MATLAB in complex arithmetic.

If you glance at the table of contents, you will see how MATLAB can be your companion in such staples of complex variable theory as conformal mapping, infinite series, contour integration, and Laplace and Fourier transforms. Fractals, the most recent interesting topic involving complex variables, cries out to be treated with a language such as MATLAB, and you might want to begin this book at its end, The Coda, which is devoted entirely to this visually intriguing subject. However, I must add that as you progress through the book, the MATLAB skill required increases gradually, and leaping to the end is not for everyone.

Sometimes while working with MATLAB and complex algebra, you may be puzzled at what you find. For example, if you ask a class of high school students to compute $\left(5^3\right)^{1/3}$, someone will quickly call out "five." He or she will know to multiply the exponents together. Now asking the same class for $\left((-1+i)^2\right)^{1/2}$, a student will follow the same logic and produce a correct answer: $-1 + i$. But another student, using MATLAB, will say $1 - i$. Both answers are correct, but why did MATLAB choose this one? This book answers such questions and many similar ones.

This textbook does not purport to present MATLAB as a substitute for a knowledge of the functions of a complex variable any more than MATLAB can be used as a replacement for an actual understanding of elementary calculus or linear algebra. This is also not a text from which one learns the elements of MATLAB, although if you already know a little of the language, it will expand your knowledge. Some books that will get the reader started in the elements of MATLAB programming are listed as references [1,2]. MATLAB is not without constraints, assumptions, limitations, irritations, and quirks, and there are subtleties involved in performing the calculus of complex variable theory with this language that will be made evident here. Without knowledge of these subtleties, the engineer or scientist who is attempting to use MATLAB for solutions of practical problems in complex variable theory suffers the real risk of making major mistakes. This book should serve as an early warning system about these pitfalls.

This book should be read as a companion to standard texts on functions of a complex variable. Throughout what follows, we refer to two of the author's favorites. Not surprisingly, one volume is his own *Complex Variables with Applications* (3rd edition) published by Addison-Wesley in 2005. We refer to this book as "W" in the text. In most cases, the section numbers referred to apply to the second edition as well. There is a Spanish translation available for those who prefer that language: *Variable Compleja con Aplicaciones*. The reference book in the Schaum's outline series, *Complex Variables*, 2nd edition by M. Spiegel et al. is an old favorite of mine, and although it is more handbook than textbook, it is remarkably well done, and I will refer to it with the letter "S." Notice that the section numbers that I refer to apply to the second edition only. *Complex Variables and Applications* by R. V. Churchill and colleagues is very well written, but the book has had so many editions that it is difficult to refer the student to any particular section.

References

1. Hunt, B., Lipsman, R., and Rosenberg, J. *A Guide to MATLAB for Beginners and Experienced Users*, 3rd edition. New York: Cambridge University Press, 2014.
2. Hahn, B. and Valentine, D. *Essential MATLAB for Engineers and Scientists*, 5th edition. New York: Academic Press, 2013.

A. David Wunsch
University of Massachusetts Lowell
Lowell, Massachusetts

A Note to the Reader

To keep the cost of this book reasonable, all figures are rendered in black and white and shades of gray. Some of the codes provided here will produce only these colors, while other programs will produce color plots on your screen even though the plots are in grayscale in your book. In the solutions manual I have not avoided color.

MATLAB® is upgraded at least once a year with new releases. Its capabilities change and for the most part are improved. This book is based on R2015a, the release available through most of the year 2015.

Corrections to this book as well as to the solutions manual will be posted at the author's web page hosted by the University of Massachusetts Lowell. Here is the URL: http://faculty.uml.edu/awunsch/wunsch_complex_variables/faculty.htm.

The author invites corrections and comments for his work. An email address can be found at the above website where you will also find solutions to the odd numbered problems. To assist readers with data entry in MATLAB, code text can be found for download at https://www.crcpress.com/A-MatLab-Companion-to-Complex-Variables/Wunsch/9781498755672.

Instructors using my book in the classroom can also receive a copy of a complete solutions manual, for all problems, if they write to me on their college stationary at the ECE Department, University of Massachusetts Lowell, Lowell, MA 01854, USA.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

1

Complex Arithmetic

1.1 The Rectangular Form

A complex number z can be stated in the form

$$z = x + iy \tag{1.1}$$

where x and y are real numbers, and there is multiplication between the i and the y. An equally valid representation is

$$z = x + yi \tag{1.2}$$

since complex numbers obey the commutative law of multiplication so that the order of multiplying y with i is immaterial. Right away we must deal with an idiosyncrasy of MATLAB®. The statement $z = 3 + 4i$ can be entered in the MATLAB command window with the result given in the following example. Note that >> preceding a line of code indicates that the expression was *entered from the keyboard* into the command window of MATLAB.

Example 1.1

```
>> z=3 + 4i
z = 3.0000 + 4.0000i
```

which is exactly what we hoped for: MATLAB has returned 3.0000 + 4.0000i. However, entering $z = 3 + i4$ in the command window will result in an error message. The value i must appear as the second factor in the multiplication if i is to be interpreted by MATLAB as a multiplicative factor in $z = x + y$i. The practice is only valid if i is *preceded* by an explicitly stated *real* (not complex) number. A symbol cannot be used for that number. This entire convention can be overlooked provided we employ the MATLAB $*$ for multiplication. We will adopt that practice throughout this book even though the authors of MATLAB claim that eliminating $*$ where allowed will speed up calculations; the advantage is often slight.

Note that in general, all the arithmetical operations that one does with real numbers in MATLAB can be carried out for complex numbers, using the same operators (i.e., the **+** and **–** signs for addition and subtraction, the / for division, the ^ for raising a number to a power, and as noted, the ∗ for multiplication). The precise meaning of what the ^ will yield when followed by a fraction will be treated in section 1.3.

To get the magnitude (or absolute value or modulus) of a complex number, we apply the operation **abs** to that number. Thus,

```
>> abs(3 + 4*i)
ans = 5
```

Example 1.2

Multiply $3 + i4$ by $(1 + 2i)$ and add $i2$ to that result. Then find the absolute value of that quantity.

Solution:

```
>> (3 + i*4)*(1 + 2*i) + 2*i
ans = –5.0000 + 12.0000i
>> abs(ans)
ans = 13
```

Electrical engineers often prefer to use j instead of i, and this notation can be used automatically in MATLAB. We just use j instead of i and follow the same conventions as above.

You should not use both i and j in the same code, or you will confuse yourself and whoever reads what you wrote.

Example 1.3

Use j instead of i, and raise $(1 + j3)$ to the –2 power, divide that result by $2 – j3$, and subtract $3 + 4j$ from that result.

Solution:

```
>> (1 + 3*j)^(–2)/(2–j*3)–(3 + 4*j)
ans = –2.9985 – 4.0277i
```

Note that although we used j instead of i, MATLAB returned an answer employing i.

For some purposes, it is more convenient to put $j = -i$, and you should make this statement at the start of your work if that is your preference. (Note that $j^2 = -1$ as before.) Recall that physicists prefer $e^{-i\omega t}$ in lieu of the electrical engineer's $e^{j\omega t}$, where ω is a radian frequency and t is the time.

The real and imaginary parts of a complex number, for example z, are found from the functions **real**(z), **imag**(z), and the conjugate of z is given by

conj(z) as shown by the following sequence of calculations. Note that in the text material of this book, we will designate the conjugate of a quantity by an overbar, as in $\bar{z}$, which in MATLAB would of course be **conj**(z).

Example 1.4

```
>> z=3 + 4*i
z = 3.0000 + 4.0000i
>> x=real(z)
x = 3
>> y=imag(z)
y = 4
>> w=conj(z)
w = 3.0000 – 4.0000i
```

1.1.1 A Caveat on Complex Numbers as Matrix Elements

MATLAB, as its name might suggest, is a computer language for technical computation that is based on matrices (i.e., rectangular arrays of numbers). Sometimes the matrix consists of just a horizontal row of numbers, in a specific order, or a column of numbers in a certain order. Such matrices are known as vectors and more specifically as row or column vectors. The numbers that make up matrices, the elements, can be complex or real numbers, or symbols. When entering complex numbers as the elements, it is useful to state them with parentheses surrounding each complex number. In other words, instead of entering 3 + 4∗i, you should enter (3 + 4∗i). Otherwise, an inadvertent use of the space bar can result in an error, as shown by the following where we want to enter a row vector a having two elements, one being $3 + i$ and the other $2–i$. In the first instance, we will do this correctly without parentheses, then incorrectly without parentheses, because we have placed a space where it does not belong, and finally correctly with parentheses even though there is an unnecessary space:

```
>> a=[3 + i   2–i]
a = 3.0000 + 1.0000i   2.0000 – 1.0000i
>> a=[3   + i   2–i]
a = 3.0000   0 + 1.0000i   2.0000 – 1.0000i
>> a=[(3   + i)   (2–i)]
a = 3.0000 + 1.0000i   2.0000 – 1.0000i
```

Comment: Note that in the second instance, the value obtained for a is

3.0000   0 + 1.0000i   2.0000 – 1.0000i

which is a row vector with *three* elements, not the desired two elements. The first element is 3, and the second is 0 + 1.0000i=i. The error was caused by our having a space after the 3 and before the adjacent + sign.

In the third output for a, we got the correct value even though we typed a space between the 3 and the plus sign following it.

Note that you can always tell the number of elements in a row or column vector, let us call it a, with the command **length**(a). If we apply this operation to the three values of a described above, we get 2, 3, and 2, respectively.

Exercises

1. Using MATLAB, determine

   a. $\dfrac{3-4i}{1+i2}+i2$

   b. $\dfrac{2+i}{3+5i}-1-i/2$

   c. $(1+i2/3)^3-\dfrac{i}{3-4i}$

   d. $\left(1+\dfrac{2}{i3}\right)^{11}$

   e. $\left(\dfrac{2i}{(3+4i)}+\overline{\left(\dfrac{1}{5-4i}\right)}\right)^5$

2. The numerical value of the expression $(1 + i)^{37}$ can be determined without recourse to MATLAB by the following technique: $(1+i)^{37}=(1+i)^{36}(1+i)=\left((1+i)^2\right)^{18}(1+i)$. Now complete the calculation by evaluating $(1 + i)^2$ and noting that i raised to any even power is easily computed. Check your answer by using MATLAB to find $(1 + i)^{37}$ directly.

3. Consider the finite series ($N \geq 0$ is an integer) and its sum.
   $\sum_{n=0}^{N} z^n = \dfrac{1-z^{N+1}}{1-z}$ $z \neq 1$, whose derivation is identical to that given in real calculus for the sum of a finite geometric series.

   a. Using MATLAB, sum the series $1+\dfrac{i}{2}+\left(\dfrac{i}{2}\right)^2+\left(\dfrac{i}{2}\right)^3+\ldots+\left(\dfrac{i}{2}\right)^{10}$ directly by using a loop created from the **while** or **for** commands. Use the "long format" of MATLAB as you will need it in part (b).

   b. Verify the answer obtained in (a) by using the closed-form expression for the sum given above, again using the long format so that you can notice any small discrepancy.

1.2 The Polar Form of Complex Numbers

Referring to Figure 1.1, we are reminded that a complex number z can be expressed using the polar coordinates r and θ. Thus,

$$z = x + iy = r\cos\theta + ir\sin\theta = r(\cos\theta + i\sin\theta) \tag{1.3}$$

Sometimes $cis\theta$ is used to mean $\cos\theta + i\sin\theta$, but other times, the notation $\angle\theta$ is used for the same quantity. Thus,

$$z = rcis\theta = r\angle\theta \tag{1.4}$$

Generally, θ is expressed in radians, but one might also use degrees.

Now $|z| = \sqrt{x^2 + y^2} = r$, i.e., the polar coordinate r is simply the magnitude of the complex number z, and we have in MATLAB that $r = \text{abs}(z)$. The quantity θ is called the *angle* or the *argument* of the complex number z.

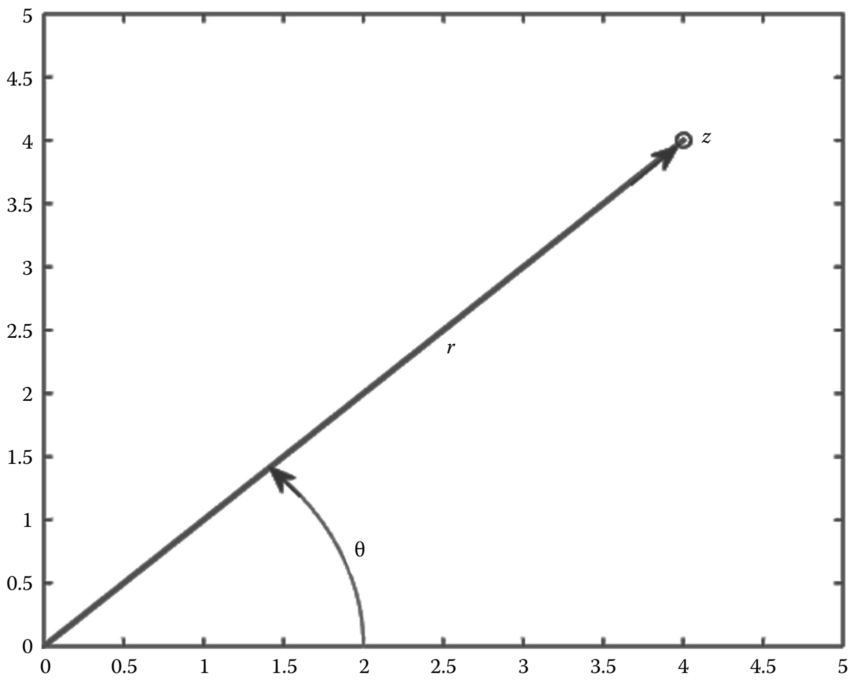


FIGURE 1.1
The polar coordinates r and θ for the point z.

There are some subtleties with θ since it is multivalued. If θ_0 is a possible value of θ, then so is $\theta = \theta_0 + 2k\pi$, where $k = 0, \pm1, \pm2, \ldots$. If we are working in degrees, the expression becomes $\theta = \theta_0 + k360°$. Given a complex number in rectangular form, MATLAB can be used to yield a value of θ in radians from the function **angle**(*z*). But which value will it yield? It yields the *principal value of the argument*, the one satisfying $-\pi < \theta \leq \pi$. Note that a few textbooks use a different definition for the principal value and that the value returned by MATLAB would not be in conformity with theirs.*

Example 1.5

Find the polar form of the complex number $-\sqrt{3} - i$.

Solution:

```
>> z= −sqrt(3)−i;
r=abs(z)
r = 2.0000
>> theta=angle(z)
theta = −2.6180
```

Thus, we have shown $z = 2\angle -2.6180\ldots$ in polar form.

A short calculation with MATLAB shows that −2.6180 agrees with −5π/6 (or −5pi/6 in MATLAB) to as many decimal places as MATLAB will display in the standard format. Construction of the vector for $-\sqrt{3} - i$ on a piece of paper and recognizing the resulting triangle with angles 30, 60, and 90 degrees should show where the angle −5π/6 arises from.

Incidentally, the above rectangular to polar calculation can be done a little more easily in MATLAB as [*th*,*r*] = **car2pol**(*x*,*y*), which converts complex numbers from Cartesian to polar form. Thus, using numbers from above,

```
>> [th,r]=cart2pol(−sqrt(3),−1)
th = −2.6180
r = 2.0000
```

which agree with what we found.

Example 1.6

Using MATLAB, compute the angles of the complex numbers −1+eps∗i and −1−eps∗i and compare them by computing their difference. Recall that in MATLAB, eps is the smallest possible difference between two floating point numbers. In the version of MATLAB used in this book, eps ≈ 2.22×10^{-16}. If you type the word eps in the MATLAB command window, you can find out what eps is for your version of MATLAB. We chose here to work in the "long" format.

* These books take $0 \leq \theta < 2\pi$.

Solution:

```
>> format long
>> angle1=angle(–1 + i*eps)
angle1 = 3.141592653589793
>> angle2=angle(–1–i*eps)
angle2 = –3.14159265358979
>> angle1–angle2
ans = 6.28318530717959
>> 2*pi
ans = 6.28318530717959
```

Note that although graphical representation of $-1 + i*\text{eps}$ and $-1 - i*\text{eps}$ as points in the complex plane will involve neighboring points that are *exceedingly* close to one another, these points lie just above and below the negative real axes. Their principal angles are nearly π and $-\pi$, respectively. The differences in their angles, as computed by MATLAB, are indistinguishable from 2π.

Referring to Figure 1.1, we see that $\theta = \arctan(y/x)$, and we might be tempted to use the function **atan** in MATLAB, which yields the arctangent, in order to obtain θ, the angle of the complex number $z = x + iy$. This is to be avoided; because of the multiplicity of values of the arctangent, the computer might yield the wrong value. If, for example, $x = 1$ and $y = 1$, then **atan**(1/1) is evaluated by MATLAB to be 0.78539816339745 or $\pi/4$, that is a correct value as a sketch of the vector for this complex number shows. However, if $x = -1$, $y = -1$, we have from MATLAB that atan(–1/–1) is again 0.78539816339745, which is incorrect for the angle of this complex number. A correct value, as a sketch of the corresponding vector shows, is $-3\pi/4 = -2.35619449019234$. In general, it is better to use the **angle** function of MATLAB to avoid these situations. However, one might use the function **atan2(*x*,*y*)** in MATLAB where the real and imaginary parts of the complex number $z = x + iy$ are entered in a particular order into the function. Additionally, the MATLAB function [*th*,*r*] = car2pol(x,y) will always yield the correct angle *th* (or θ).

Given a complex number in the polar form (a magnitude r and an angle θ), we can compute the Cartesian form by asking MATLAB for $r\cos\theta = x$ and $r\sin\theta = y$. Alternatively, we can use a function M file built into MATLAB that will automatically do the two calculations. The function is [*x*,*y*] = **pol2cart**(*th*,*r*), where the angle θ in radians, here called *th*, is entered first followed by *r*. Suppose we wish to convert the complex number $2\angle\pi/6$ to rectangular form. We proceed as follows:

```
>> [x,y] = pol2cart(pi/6,2)
x = 1.73205080756888
y = 1.00000000000000
```

Note that the program does not yield the complex number $z = x + iy$. It simply gives the real and imaginary parts of z. To get z, one must take the two results from above and make that additional computation.

Exercises

1. We know that the polar angle of the product of two complex numbers can be expressed as the sum of the angles of each.
   a. Multiply $(1 + i)$ by $\left(\sqrt{3}+i\right)$ using MATLAB. Find the angle of the product using MATLAB and show that it is the sum of the angles of each factor.
   b. Repeat the above exercise, but use as factors $(1 + i)$ and $-\sqrt{3}+i$. Explain why the angle of each factor, obtained from MATLAB, when added, does not agree with the angle that MATLAB yields for the angle of the product. Compute these angles using MATLAB.
2. When MATLAB is asked for the argument (angle) of a complex number, it always returns the principal value. Use MATLAB to find the principal argument of the following numbers where all angles are in radians except in part (d).
   a. $5cis(197)$
   b. $1/(4\angle 79)$
   c. $(1 + 2i)^{17}$
   d. $(5cis(53.3°))^{8}$ (give answer in degrees)
3. Using the **pol2cart** function in MATLAB described above, convert these numbers to rectangular components where the angles below are in radians:
   a. $3cis(-207)$
   b. $4\angle 50$
4. Write a new function, similar to the **pol2cart** function in such a way that if it is given θ in *degrees* and r, it will convert the polar form of the number to rectangular form. Call the new function **pold2cart**. Use it to convert these numbers to rectangular form. Give x and y.
   a. $5\angle 60°$
   b. $2\angle(-320°)$

1.3 Fractional Powers of Complex Numbers

First, we make a note on notation. In this book, as in many texts on complex variable theory, the symbol $\sqrt{\beta}$ is applied only when β is a non-negative real number and is intended to mean the non-negative square root. Thus, $\sqrt{16}$ is always 4 and never −4. Similarly, the $\sqrt[m]{\beta}$ is applied when β has this same restriction and m is an integer and refers to the mth root of β but only to the non-negative real value. Thus, $\sqrt[3]{64}$ is 4 and not any of the other two possible values, which are complex.

Suppose you wish to solve the equation $z^4 - 1 = 0$. We have $z^4 = 1$. Notice that our equation has four solutions: ±1 and $\pm i$. However, if you attempt to solve the given equation using MATLAB by asking it for $z = 1^{1/4}$, which is written in MATLAB as 1^(1/4), you will receive just one of the four answers, in this case, 1. The question arises as to how MATLAB decides which answer to give you when a number is raised to a fractional power and how we might get the rest.

If z is to be raised to a fractional power n/m where m and n are integers, with $m \neq 0$, and where the fraction n/m cannot be reduced, we convert z to the polar form $z = r\angle\theta$, and as shown in the standard texts, we have m values given by

$$z^{n/m} = \left(\sqrt[m]{r}\right)^n cis\left[\frac{n}{m}\theta + 2k\pi\frac{n}{m}\right] \quad k = 0, 1, 2, \ldots |m| - 1 \tag{1.5}$$

where as we recall, $cis\phi = \cos\phi + i\sin\phi$. This result is consistent with the definition $z^{n/m} = (z^{1/m})^n$. Notice the order of the operations.

In what follows, we will take m as positive and skip the absolute magnitude signs around the m. We can do this because it is easily shown that the set of values of $z^{-n/m}$ is identical to the set of values of $z^{n/(-m)}$. Thus, given the problem of computing $(1 + i)^{2/(-3)}$, we can compute the set of values of $(1 + i)^{-2/3}$ and use a positive m.

Equation 1.5 is equivalent to

$$z^{n/m} = \left(\sqrt[m]{r}\right)^n cis\left[\frac{n}{m}\theta\right] cis\left[2k\pi\frac{n}{m}\right] \quad k = 0, 1, 2, \ldots (m - 1) \tag{1.6}$$

which arises from the identity $cis(\alpha + \beta) = cis\alpha cis\beta$; it applies to any numbers α and β. Note that in Equation 1.6, we can generate the same set of values for $z^{n/m}$ if we let k go through any set of m consecutive integers, for example, $1, 2, \ldots m$.

The results from Equations 1.5 and 1.6 are m numerically distinct values. When plotted as points in the complex plane, they all lie on a circle having radius $\left(\sqrt[n]{r}\right)^m$ (i.e., they have identical magnitudes). The angular spacing of these values is $2\pi/m$. The separation of these points on the circle is uniform.

If you ask MATLAB for the computation of a fractional power, for example, $(81i)^{3/4}$, you will receive exactly one value, in this case, 10.3325+24.9447i, which in polar form is 27 $\angle 1.1781\ldots$. The angle here is $3\pi/8$ radians. According to Equations 1.5 and 1.6, there are three other values for $(81i)^{3/4}$. After some experimentation with MATLAB, you will see how it decides which value of $z^{n/m}$ to give you.

MATLAB Rule for Fractional Powers *The angle* θ *in Equations 1.5 and 1.6 is taken as the principal value, and the value of k is set equal to zero.* MATLAB *returns the result in Cartesian form.*

You should confirm that this convention yields the MATLAB result just given for $(81i)^{3/4}$.

Example 1.7

Use Equation 1.5, taking the MATLAB convention for fractional powers, to determine one value of $(-1-i)^{2/3}$, and verify that MATLAB does indeed yield this value.

Solution:

We have $-1-i=\sqrt{2}\angle(-3\pi/4)$, where the principal value of the argument of $(-1-i)$ is used.

Using Equation 1.5, with $k = 0$ we have

$$(-1-i)^{2/3}=\left[\sqrt{2}\angle(-3\pi/4)\right]^{2/3}=\left(\sqrt[3]{\sqrt{2}}\right)^{2}\angle\left(-3\frac{\pi}{4}\frac{2}{3}\right)=\sqrt[3]{2}\angle\left(-\frac{\pi}{2}\right)=-i\sqrt[3]{2}$$

Proceeding with MATLAB, we have

```
>> (-1-i)^(2/3)
ans = 0.0000 - 1.2599i
```

These two answers are numerically identical because the cube root of 2 is 1.2599….

If we compute a value of $z^{n/m}$ and raise it to the m/n power, do we necessarily get back z? That is, does $(z^{n/m})^{m/n}$ equal z when the inner and outer operations are done in MATLAB? Because MATLAB will choose just one value of the n possible values when raising a number to the m/n power, you cannot say for certain that in MATLAB $(z^{n/m})^{m/n} = z$.

The reader should confirm that in MATLAB we have $\left((-1)^{4/3}\right)^{3/4}=-i$, so the original z (namely, -1) is *not* returned. The result comes about because the principal angle of -1 is π. When MATLAB evaluates $(-1)^{4/3}$ using this angle, and the rule stated above, the result is $\frac{-1}{2}-i\frac{\sqrt{3}}{2}$ whose polar representation, using the principal angle, is $1\angle\left(-2\frac{\pi}{3}\right)$. Raising the preceding to the ¾ power using the MATLAB rule for fractional powers, we get $1\angle\left(-\frac{\pi}{2}\right)=-i$.

For a different problem, MATLAB yields $\left((-i)^{4/3}\right)^{3/4} = -i$. The value of z is returned. One should be able to predict in advance whether the original z is obtained if we compute $(z^{n/m})^{m/n}$. This matter is explored in problem 2 in the Exercises.

1.3.1 Using sqrt

If you want to obtain the square root of a number, z, using MATLAB, you could ask for the computation z^(1/2). The resulting root is the one supplied by the MATLAB rule given above for fractional powers. Alternatively, we might ask for **sqrt**(z), which requires about the same amount of typing. The same root is obtained as if you were using z^(1/2)—that is, you get $\sqrt{r}\,cis(\theta/2)$, where θ is the principal angle of z. So, for example, we have

```
>> format short
>> sqrt(–9)
ans = 0 + 3.0000i
```

We did not get –3i, the other square root.

1.3.2 Reminder: A Warning about Square Roots and Fractional Powers in MATLAB

If z is a matrix and you ask MATLAB for **sqrt**(z), then MATLAB will return to you a matrix whose elements are the square root of each element of z. However, if you ask MATLAB for z^(1/2), it will return the square root of the matrix z (provided it is square matrix, like 2×2)—that is, a matrix whose square is z. To get the same matrix as produced by **sqrt**(z), you will need z.^(1/2), where a dot follows the z. Similar precautions must be taken where other exponents are concerned.

1.3.3 Use of roots

Suppose we want to obtain all m values of $z^{n/m}$ through the use of MATLAB. A simple method involves the function **roots**. Let us begin by using **roots** to find $z^{1/m}$, where m is a positive integer. Observe that all m solutions for the unknown w in the equation $w^m = z$ are the m values of $z^{1/m}$. MATLAB has a function called **roots** that will solve the polynomial equation

$$c_1 w^m + c_2 w^{m-1} + c_3 w^{m-2} + \ldots + c_m w + c_{m+1} = 0$$

and yield all m roots. The procedure is to ask MATLAB for **roots(p)**, where **p** is a row vector whose entries, from left to right, are the numerical coefficients $c_1, c_2, \ldots c_{m+1}$. Note that if we take $c_1 = 1, c_2 = 0, c_3 = 0, \ldots c_{m+1} = -z$ in the polynomial equation, we obtain $w^m - z = 0$ whose solutions all satisfy $w = z^{1/m}$.

Suppose for example we wanted all six values of $(1 + i)^{1/6}$. This is the same problem as finding all six solutions of the polynomial equation $w^6 - (1 + i) = 0$. The expression on the left is a polynomial of degree 6. We should see that this polynomial has coefficients of zero for the terms w^5, w^4, …, w, while the coefficient of w^0 is $-(1 + i)$. Thus, to solve our polynomial sixth-degree equation in w, we proceed as follows:

```
>> coeffs = [1 0 0 0 0 0 −(1 + i)];
>> format short
>> roots(coeffs)
ans =
  −1.0504 − 0.1383i
  −0.6450 + 0.8405i
  −0.4054 − 0.9788i
   0.4054 + 0.9788i
   0.6450 − 0.8405i
   1.0504 + 0.1383i
```

The above are the six values of $(1 + i)^{1/6}$. The following provides a check.

```
  >> ans.^6
ans =
  1.0000 + 1.0000i
  1.0000 + 1.0000i
  1.0000 + 1.0000i
  1.0000 + 1.0000i
  1.0000 + 1.0000i
  1.0000 + 1.0000i
```

There is some round-off error, which is hidden because we are using short format, but each of these expressions is, to our degree of approximation, (1 + i). This is correct, as we expect all six values of $(1 + i)^{1/6}$ when raised to the sixth power to yield (1 + i).

Suppose we want all seven values of $(3 + 4i)^{5/7}$. We would proceed as before, calculating $(3 + 4i)^{1/7}$. We would then take all seven of these values and raise them to the fifth power. Here is the procedure:

```
>> coeffs = zeros(1,8);
%the above takes all the coeffs as zero at first
coeffs(1) = 1;
coeffs(8) = −(3 + 4*i);
poly_roots = roots(coeffs);
final_answer = poly_roots.^5
```

The output is

```
final_answer =
  -3.0852 - 0.6691i
   0.0342 + 3.1567i
   3.0700 - 0.7358i
  -1.4005 - 2.8293i
   1.3388 - 2.8590i
  -2.4467 + 1.9950i
   2.4894 + 1.9414i
```

Note that there are seven answers, as expected. The first line of our code coeffs=zeros(1,8); establishes a row vector having eight elements. Initially all eight are zero. Then we set the first element equal to one and the last equal to $-(3 + 4i)$.

The above technique can be applied to finding all $|m|$ values of $z^{n/m}$ when m is a *negative* integer. We compute $z^{-n/|m|}$ and proceed as above, first finding the $|m|$ values of $z^{1/|m|}$ and then raising them to the $-n$th power.

Suppose we were to take each of the values of $(3 + 4i)^{5/7}$ obtained above and, using MATLAB, raise them all to the 7/5 power. Can we expect in each case to recover $(3 + 4i)$? Trying this, we have

```
>> recover = final_answer.^(7/5)
recover =
  -2.8772 + 4.0892i
  -2.8772 + 4.0892i
   4.7313 - 1.6171i
  -4.7782 - 1.4727i
  -0.0759 - 4.9994i
  -4.7782 - 1.4727i
   3.0000 + 4.0000i
```

Notice that only one of these results is the original $3 + 4i$. This is because there are five possible values of a number raised to the 7/5th power. Just one of the above yields ultimately to our having $\left((3+4i)^{5/7}\right)^{7/5} = 3+4i$.

Keep the following in mind: Suppose using any method at your disposal you compute *all* $|m|$ values of $z^{n/m}$, where n and m are integers, and $m \neq 0$. Now suppose you raise each value to the m/n power using the MATLAB expression V^(m/n), where V is any one of those m values of $z^{n/m}$. There is no guarantee that you will get back z. A simple example will suffice. We know that $(-1)^{3/2} = (-1)^{1/2}\,(-1) = \pm i$. From MATLAB, we raise both i and $-i$ to the 2/3rds power:

```
>> i^(2/3)
ans = 0.5000 + 0.8660i
```

```
>> (–i)^(2/3)
ans = 0.5000 – 0.8660i
```

In neither case did we recover –1. However, if you computed *all* values of $i^{2/3}$ and $(-i)^{2/3}$, you would find in each case that one of the values obtained is –1.

1.3.4 A Program to Give You Fractional Roots and to Do a Check

Here is a simple MATLAB program that asks you to choose a value of z and a positive integer m. The program then gives you all values of $z^{1/m}$. It does this by solving the polynomial equation in w : $w^m - z = 0$ using **roots**.

We have added some additional checks to our program. Notice that if you were to factor $w^m - z$ into its m factors $(w - w_1)(w - w_2) \ldots (w - w_m)$, then the product of the roots $w_1 w_2 \ldots w_m$ is seen to equal $(-1)^{m-1}z$; the product of the m values of $z^{1/m}$ is thus $(-1)^{m-1}z$. Similarly, you should convince yourself that the coefficient of w^{m-1} obtained when you multiply the m factors is $-(w_1 + w_2 + w_3 + \ldots w_m)$. But since w^{m-1} does not appear in $w^m - z$, we conclude that the sum of the m values of $z^{1/m}$ is zero.

Try the following problem with some numbers:

```
z=input('the complex value of z to be considered')
m=input('input the positive integer m')
v=zeros(1,m+1);%creates a row vector length m+1
%elements are zero.
v(1,1)=1;% the first coeff is 1, the coeff of w^m
v(1,m+1)=-z;% the constant term (the last coeff) is -z
%the following solves the equation w^m-z=0 for w=z^(1/m)
the_roots =roots(v)
prod_roots=prod(the_roots)
%the preceding should equal (-1)^(m-1) times z.
check_prod=(-1)^(m-1)*z
check_roots=sum(the_roots)
%the preceding should be zero to a good approx.
```

As an exercise, the reader should modify the above code so that the user is prompted to enter an integer n and a positive integer m where the fraction n/m is irreducible. The program then computes all m values of $z^{n/m}$ where again z must be supplied.

1.3.5 A Further Caveat with Fractional Powers: The Plot Function

Suppose x is a row vector whose elements are real numbers. Let $w(x)$ be a function defined for these values of x. A commonly used line in MATLAB code is **plot**(x,w), which will produce a plot in the Cartesian plane of w versus x. For example, if **x = linspace**(0,10,1000) and $w = \sin(x)$, the command **plot** (x,w) gives you a nice plot of the sine function based on 1000 data points spaced between 0 and 10.

One must be careful, however. If w is a complex function of the real variable x, then the command plot (x,w) yields a plot of *only the real part* of w versus x. This is demonstrated in the following example.

Example 1.8

Using the MATLAB **plot** function, graph $w = x^{1/3}$ for the interval $-2 \le x \le 2$ and explain your result. Use at least 100 data points.

Solution:

```
x = linspace (-2,2,101);
w = x.^(1/3);
plot(x,w);grid
```

which results in Figure 1.2.

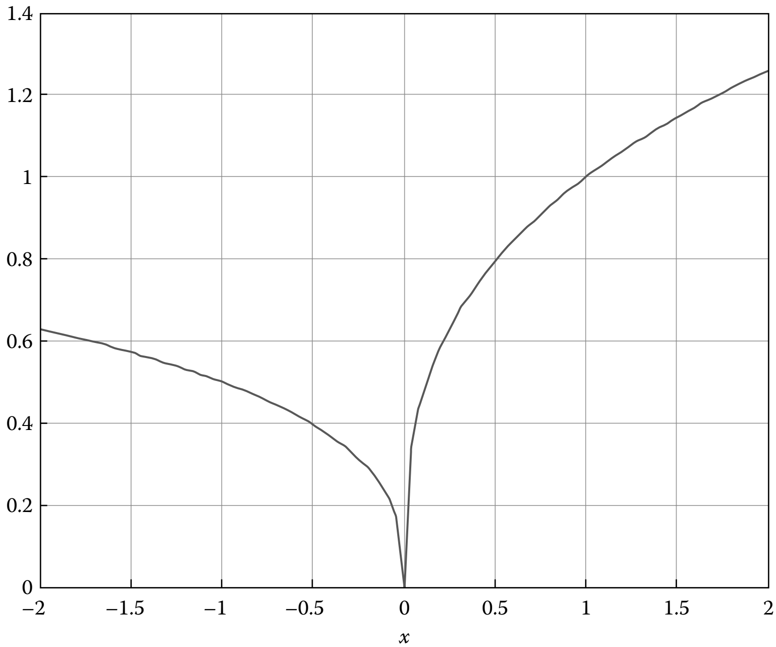


FIGURE 1.2
The plot for Example 1.8.

Notice the lack of symmetry about $x = 0$. In executing the program, MATLAB issues a warning about the imaginary value of the function being ignored in the plotting process.

To explain this result, notice that

$$x^{1/3} = \sqrt[3]{|x|}\angle\left(\frac{1}{3}\theta + \frac{2}{3}2k\pi\right) \quad k = 0, 1, 2$$

MATLAB employs the principal value, taking $k = 0$, and $\theta = 0$ when $x > 0$. Thus, for $x \geq 0$, we have $x^{1/3} = \sqrt[3]{|x|}$. If $x < 0$, we have from the principal value

$$x^{1/3} = \sqrt[3]{|x|}\angle\left(\frac{1}{3}\pi\right)$$

because $\theta = \pi$ on the negative real axis. In Cartesian form, the preceding is

$$x^{1/3} = \sqrt[3]{|x|}(\cos(\pi/3) + \text{i}\sin(\pi/3)) = \sqrt[3]{|x|}\left(1/2 + \text{i}\ \sqrt{3}/2\right)$$

In our preceding graph above, we have the curve $\frac{1}{2}\sqrt[3]{|x|}$ for $x < 0$ because only the real part of $x^{1/3}$ is plotted by MATLAB. This is exactly half the value obtained for the corresponding positive values of x and is confirmed if we study the plot.

To properly graph $x^{1/3}$ over the given interval, we must employ both the real and imaginary parts of this function as in the following code:

```
x=linspace (-2,2,101);
w=x.^(1/3);
plot(x,real(w),'-',x,imag(w),'*')
grid
```

This results in Figure 1.3, to which we have added some labels.

Finally, as we are on the subject of plotting, the reader should be reminded that the command **plot**(w), where w is a row vector having complex elements, yields a graph in which the imaginary part of w is plotted against the real part. In other words, w is plotted as points in the complex plane. Here is a simple example that yields a circle of radius 10 in the complex plane:

```
>> theta = linspace(0,2*pi,100);
w = 10*exp(i*theta);
plot(w);axis equal
```

The reader should verify this. Note that **axis equal** ensures that a circular plot appears. Without it, MATLAB might use different scales for the two axes, and the circle would be distorted into an ellipse.

We could have also just plotted 100 stars, like this ∗, for the data points and not obtained a continuous graph if we used plot(w,'∗') instead of plot(w). Here is the result. (See Figure 1.4.)

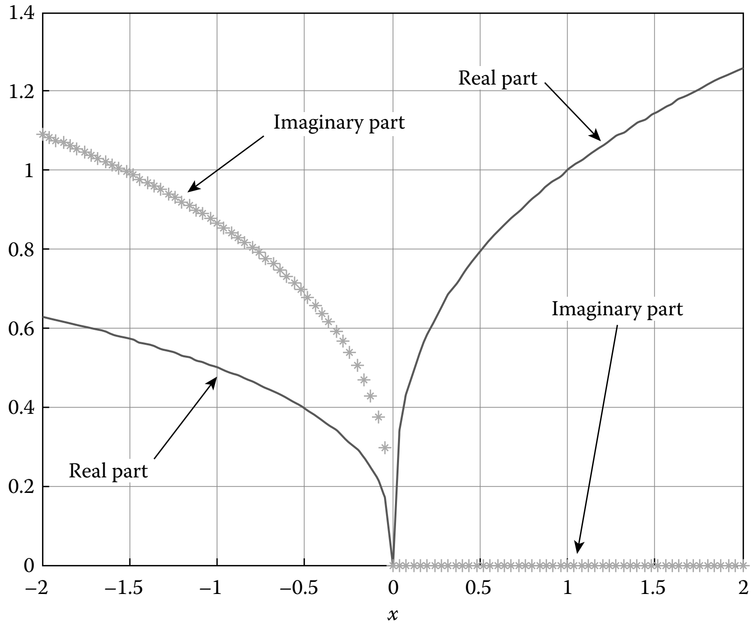


FIGURE 1.3
Plots for $x^{1/3}$.

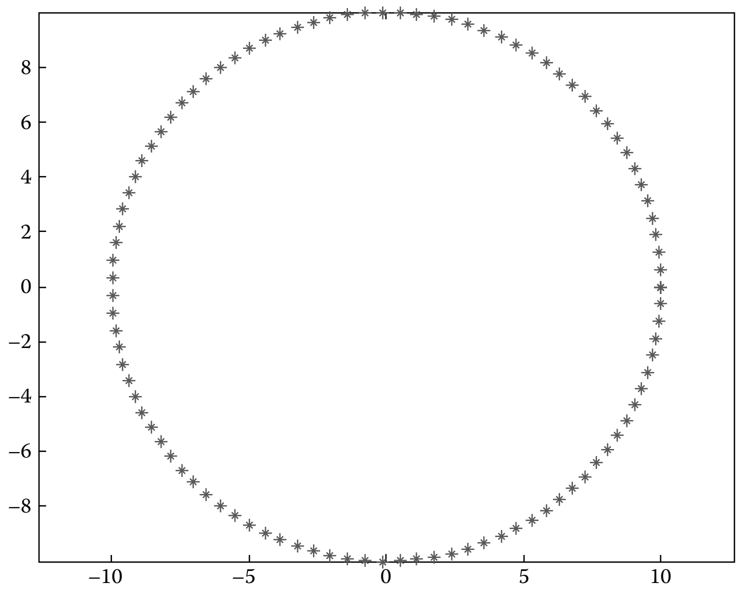


FIGURE 1.4
Values of 10eiθ generated by letting θ go through 100 values between 0 and 2π.

And while we are on the subject of **plot**, here is a further peril involving **plot** when you are using vectors with complex elements together with vectors having real elements. Consider the following:

```
clf
a=[1+i 5+5*i 10+10*i];
b=[1 5 10];
plot(a,'k-','linewidth',2);hold on
text(4,4,'the plot of a')
plot(a,'o')
plot(b,'k-','linewidth',2);
plot(b,'o');
text(2.2,5,'the plot of b');grid
```

We obtain Figure 1.5.

The plot of the vector a shows its imaginary elements plotted against their corresponding real parts. They all lie on the line $y = x$, which makes a 45-degree angle with the real axis.

Now the vector b has only real elements, which are 1 and 5 and 10. The imaginary part of each element is zero. So, following the logic used by MATLAB in making the plot for a, we would expect that these points would occur on the line $y = 0$ at $x = 1$ and 5 and 10. We would expect MATLAB to connect these points with a horizontal line.

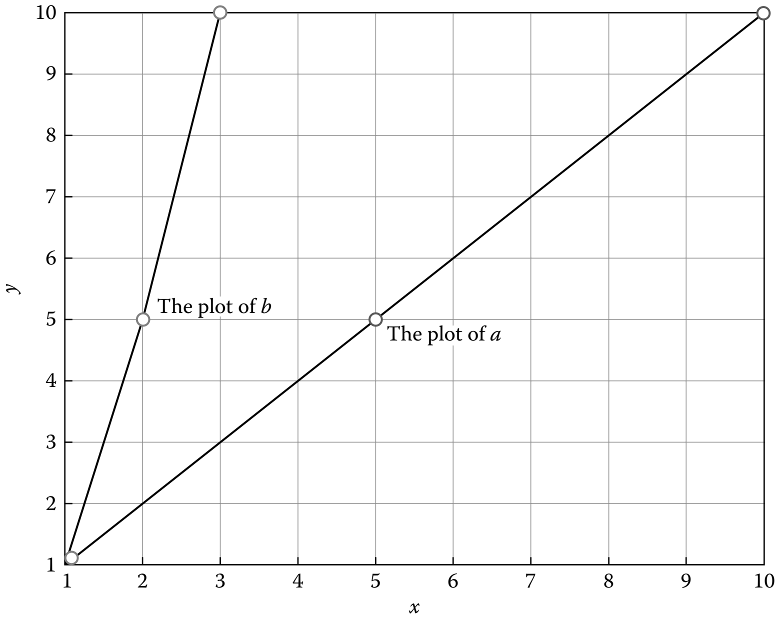


FIGURE 1.5
Perils of using the **plot** function.

Instead, MATLAB, seeing that each element in b is real, plots the value of that element against its *index* in the row vector b. Thus, the first element, one, is plotted against its index, one. The second element has the value 5 and this value is plotted against 2, and finally the third element is 10 and this is plotted at $x = 3$, $y = 10$. The preceding exercise should serve as a warning and an explanation of seemingly strange results.

Exercises

1. Using MATLAB, obtain one numerical value for each of the following by simply asking for a^b.
   a. $(-27i)^{2/3}$
   b. $(1 + i)^{1/15}$
   c. $(-1-i)^{1/9}$
2. Without using MATLAB, but knowing its convention for producing fractional roots, predict whether or not the use of MATLAB will yield these results, and explain your logic:
   a. $(i^{1/2})^2 = i$
   b. $(i^{3/2})^{2/3} = i$
   c. $((-1)^{4/3})^{3/4} = -1$
   d. $((-1)^{3/4})^{4/3} = -1$
   e. $((-1 + i)^{5/3})^{3/5} = -1 + i$
   f. Now confirm your predictions for parts (a) through (e) by using MATLAB.
3. Using **roots**, as discussed, find all values of the following and in each case plot the values using a $*$ to mark their location.
   a. $(1 + i)^{24/5}$
   b. $(3 - 4i)^{5/7}$
   c. $(13 + 7i)^{100/99}$ (be *sure* to use the function **zeros** in this case to create the coefficients for **roots**)
4. Using Equation 1.6, we can sum all the m possible values of $z^{n/m}$. The result is

$$S = \left(\sqrt[m]{r}\right)^n cis\left[\frac{n}{m}\theta\right]\sum_{k=0}^{m-1} cis\left[2k\pi\frac{n}{m}\right]$$

   We can show that this summation $S = 0$. In other words, the sum of all possible values of $z^{n/m}$ is zero.