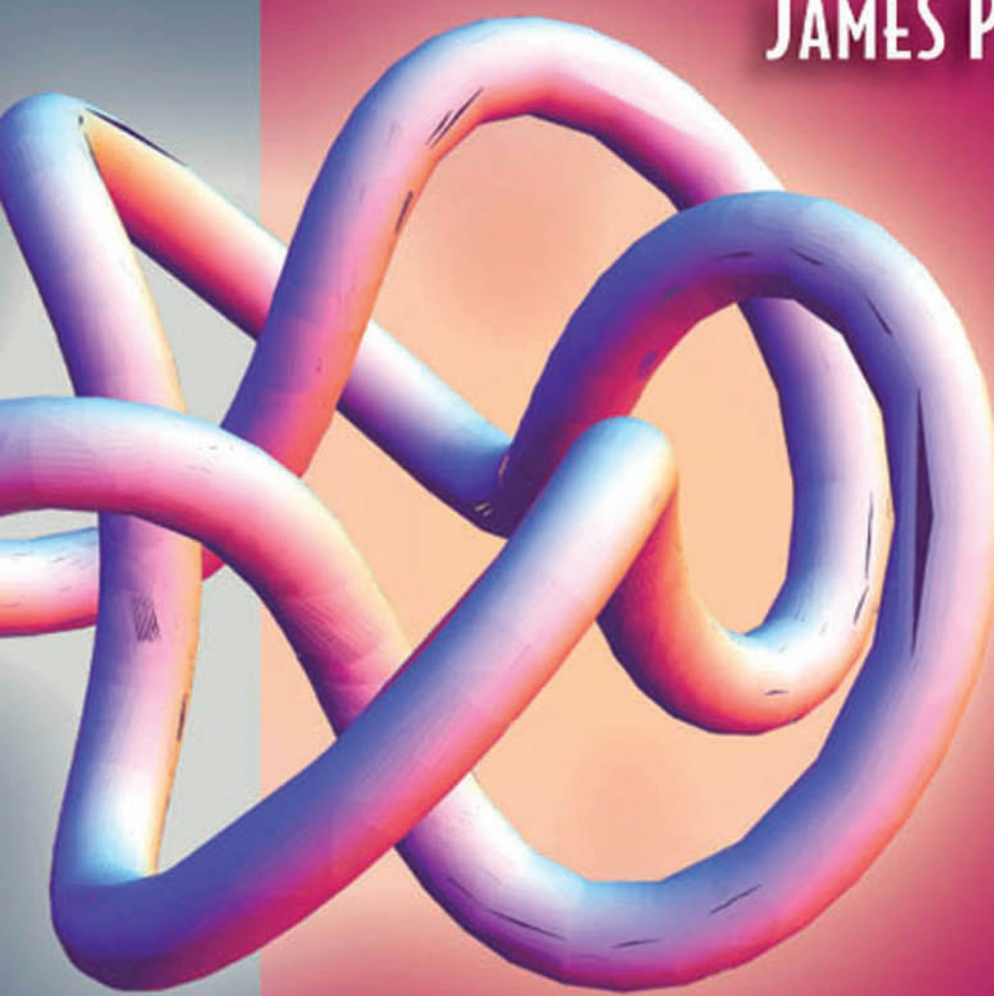


MATHEMATICA

BY **EXAMPLE** **4**^{TH EDITION}

MARTHA L. ABELL
JAMES P. BRASELTON



Wolfram
Mathematica 6



Mathematica by Example

This page intentionally left blank

Mathematica by Example

Fourth Edition

Martha L. Abell and James P. Braselton

Department of Mathematical Sciences
Georgia Southern University
Statesboro, Georgia



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Academic Press is an imprint of Elsevier



Academic Press is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
525 B Street, Suite 1900, San Diego, California 92101-4495, USA
84 Theobald's Road, London WC1X 8RR, UK

This book is printed on acid-free paper. ☺

Copyright © 2009 by Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+44) 1865 843830, fax: (+44) 1865 853333, E-mail: permissions@elsevier.com. You may also complete your request on-line via the Elsevier homepage (<http://elsevier.com>), by selecting "Support & Contact" then "Copyright and Permission" and then "Obtaining Permissions."

Library of Congress Cataloging-in-Publication Data
APPLICATION SUBMITTED

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN: 978-0-12-374318-3

For information on all Academic Press publications visit our Web site at www.books.elsevier.com
--

Printed in the United States of America
09 10 11 12 9 8 7 6 5 4 3 2 1

Working together to grow
libraries in developing countries

www.elsevier.com | www.bookaid.org | www.sabre.org

ELSEVIER

BOOK AID
International

Sabre Foundation

Contents

Preface	ix
CHAPTER 1 Getting Started	1
1.1 Introduction to Mathematica	1
<i>A Note Regarding Different Versions of Mathematica</i>	2
1.1.1 Getting Started with Mathematica	3
<i>Preview</i>	13
Five Basic Rules of Mathematica Syntax	13
1.2 Loading Packages	13
1.2.1 Packages Included with Older Versions of Mathematica	14
1.2.2 Loading New Packages	15
1.3 Getting Help from Mathematica	17
<i>Mathematica Help</i>	24
1.4 Exercises	28
CHAPTER 2 Basic Operations on Numbers, Expressions, and Functions	31
2.1 Numerical Calculations and Built-in Functions	31
2.1.1 Numerical Calculations	31
2.1.2 Built-in Constants	34
2.1.3 Built-in Functions	35
A Word of Caution	38
2.2 Expressions and Functions: Elementary Algebra	39
2.2.1 Basic Algebraic Operations on Expressions	39
2.2.2 Naming and Evaluating Expressions	44
2.2.3 Defining and Evaluating Functions	47
2.3 Graphing Functions, Expressions, and Equations	52
2.3.1 Functions of a Single Variable	52
2.3.2 Parametric and Polar Plots in Two Dimensions	65
2.3.3 Three-Dimensional and Contour Plots: Graphing Equations	71
2.3.4 Parametric Curves and Surfaces in Space	82
2.3.5 Miscellaneous Comments	94
2.4 Solving Equations	100
2.4.1 Exact Solutions of Equations	100
2.4.2 Approximate Solutions of Equations	110
2.5 Exercises	115

CHAPTER 3	Calculus	117
3.1	Limits and Continuity	117
3.1.1	Using Graphs and Tables to Predict Limits	117
3.1.2	Computing Limits	121
3.1.3	One-Sided Limits	123
3.1.4	Continuity	124
3.2	Differential Calculus	128
3.2.1	Definition of the Derivative	128
3.2.2	Calculating Derivatives	135
3.2.3	Implicit Differentiation	138
3.2.4	Tangent Lines	139
3.2.5	The First Derivative Test and Second Derivative Test	148
3.2.6	Applied Max/Min Problems	156
3.2.7	Antidifferentiation	164
3.3	Integral Calculus	168
3.3.1	Area	168
3.3.2	The Definite Integral	174
3.3.3	Approximating Definite Integrals	179
3.3.4	Area	180
3.3.5	Arc Length	186
3.3.6	Solids of Revolution	190
3.4	Series	201
3.4.1	Introduction to Sequences and Series	201
3.4.2	Convergence Tests	205
3.4.3	Alternating Series	209
3.4.4	Power Series	210
3.4.5	Taylor and Maclaurin Series	213
3.4.6	Taylor's Theorem	217
3.4.7	Other Series	220
3.5	Multivariable Calculus	221
3.5.1	Limits of Functions of Two Variables	222
3.5.2	Partial and Directional Derivatives	224
3.5.3	Iterated Integrals	238
3.6	Exercises	246
CHAPTER 4	Introduction to Lists and Tables	251
4.1	Lists and List Operations	251
4.1.1	Defining Lists	251
4.1.2	Plotting Lists of Points	258
4.2	Manipulating Lists: More on Part and Map	269
4.2.1	More on Graphing Lists: Graphing Lists of Points Using Graphics Primitives	277
4.2.2	Miscellaneous List Operations	283

4.3	Other Applications	283
4.3.1	Approximating Lists with Functions	283
4.3.2	Introduction to Fourier Series	287
4.3.3	The Mandelbrot Set and Julia Sets	299
4.4	Exercises	311

CHAPTER 5 Matrices and Vectors: Topics from Linear Algebra and Vector Calculus 317

5.1	Nested Lists: Introduction to Matrices, Vectors, and Matrix Operations	317
5.1.1	Defining Nested Lists, Matrices, and Vectors	317
5.1.2	Extracting Elements of Matrices	322
5.1.3	Basic Computations with Matrices	325
5.1.4	Basic Computations with Vectors	329
5.2	Linear Systems of Equations	337
5.2.1	Calculating Solutions of Linear Systems of Equations	337
5.2.2	Gauss–Jordan Elimination	342
5.3	Selected Topics from Linear Algebra	349
5.3.1	Fundamental Subspaces Associated with Matrices	349
5.3.2	The Gram–Schmidt Process	351
5.3.3	Linear Transformations	355
5.3.4	Eigenvalues and Eigenvectors	358
5.3.5	Jordan Canonical Form	361
5.3.6	The QR Method	364
5.4	Maxima and Minima Using Linear Programming	366
5.4.1	The Standard Form of a Linear Programming Problem	366
5.4.2	The Dual Problem	368
5.5	Selected Topics from Vector Calculus	374
5.5.1	Vector-Valued Functions	374
5.5.2	Line Integrals	384
5.5.3	Surface Integrals	387
5.5.4	A Note on Nonorientability	391
5.5.5	More on Tangents, Normals, and Curvature in $\mathcal{R}^3$	404
5.6	Matrices and Graphics	415
5.7	Exercises	430

CHAPTER 6 Applications Related to Ordinary and Partial Differential Equations 435

6.1	First-Order Differential Equations	435
6.1.1	Separable Equations	435
6.1.2	Linear Equations	442

6.1.3	Nonlinear Equations	450
6.1.4	Numerical Methods	453
6.2	Second-Order Linear Equations	457
6.2.1	Basic Theory	457
6.2.2	Constant Coefficients	458
6.2.3	Undetermined Coefficients	464
6.2.4	Variation of Parameters	470
6.3	Higher-Order Linear Equations	472
6.3.1	Basic Theory	472
6.3.2	Constant Coefficients	473
6.3.3	Undetermined Coefficients	475
6.3.4	Laplace Transform Methods	481
6.3.5	Nonlinear Higher-Order Equations	492
6.4	Systems of Equations	492
6.4.1	Linear Systems	492
6.4.2	Nonhomogeneous Linear Systems	505
6.4.3	Nonlinear Systems	511
6.5	Some Partial Differential Equations	532
6.5.1	The One-Dimensional Wave Equation	532
6.5.2	The Two-Dimensional Wave Equation	537
6.5.3	Other Partial Differential Equations	547
6.6	Exercises	550
	References	557
	Index	559

Preface

Mathematica by Example bridges the gap that exists between the very elementary handbooks available on Mathematica and those reference books written for the advanced Mathematica users. This book is an appropriate reference for all users of Mathematica and, in particular, for beginning users such as students, instructors, engineers, businesspeople, and other professionals first learning to use Mathematica. This book introduces the very basic commands and includes typical examples of applications of these commands. In addition, the text also includes commands useful in areas such as calculus, linear algebra, business mathematics, ordinary and partial differential equations, and graphics. In all cases, however, examples follow the introduction of new commands. Readers from the most elementary to advanced levels will find that the range of topics covered addresses their needs.

Taking advantage of Version 6 of Mathematica, *Mathematica by Example*, Fourth Edition, introduces the fundamental concepts of Mathematica to solve typical problems of interest to students, instructors, and scientists. The fourth edition is an extensive revision of the text. Features that make this edition easy to use as a reference and as useful as possible for the beginner include the following:

1. **Version 6 compatibility.** All examples illustrated in this book were completed using Version 6 of Mathematica. Although many computations can continue to be carried out with earlier versions of Mathematica, we have taken advantage of the new features in Version 6 as much as possible.
2. **Applications.** New applications, many of which are documented by references from a variety of fields, especially biology, physics, and engineering, are included throughout the text.
3. **Detailed table of contents.** The table of contents includes all chapter, section, and subsection headings. Along with the comprehensive index, we hope that users will be able to locate information quickly and easily.
4. **Additional examples.** We have considerably expanded the topics throughout the book. The results should be more useful to instructors, students, businesspeople, engineers, and other professionals using Mathematica on a variety of platforms. In addition, several sections have been added to make it easier for the user to locate information.

5. **Comprehensive index.** In the index, mathematical examples and applications are listed by topic or name, and commands along with frequently used options are also listed. Particular mathematical examples as well as examples illustrating how to use frequently used commands are easy to locate. In addition, commands in the index are cross-referenced with frequently used options. Functions available in the various packages are cross-referenced both by package and alphabetically.
6. **CD included.** All Mathematica code that appears in this edition is included on the CD packaged with the text.
7. **Exercises at the end of each chapter.** Each chapter of this edition concludes with a section of exercises that range from easy to difficult.

We began *Mathematica by Example* in 1990 and the first edition was published in 1991. Back then, we were on top of the world using Macintosh IIcx's with 8 megs of RAM and 40-meg hard drives. We tried to choose examples that we thought would be relevant to beginning users—typically in the context of mathematics encountered in the undergraduate curriculum. Those examples could also be carried out by Mathematica in a timely manner on a computer as powerful as a Macintosh IIcx.

Now, we are on the top of the world with iMacs with dual Intel processors complete with 2 gigs of RAM and 250-gig hard drives, which will almost certainly be obsolete by the time you read this. The examples presented in this book continue to be the ones that we think are most similar to the problems encountered by beginning users and are presented in the context of someone familiar with mathematics typically encountered by undergraduates. However, for this edition of *Mathematica by Example*, we have taken the opportunity to expand on several of our favorite examples because the machines now have the speed and power to explore them in greater detail.

Other improvements to the fourth edition include the following:

1. Throughout the text, we have attempted to eliminate redundant examples and added several interesting ones. The following changes are especially worth noting:
 - (a) In Chapter 2, we have increased the number of parametric and polar plots in two and three dimensions. For a sample, see Examples 2.3.17, 2.3.18, 2.3.21, and 2.3.23.
 - (b) In Chapter 3, we have improved many examples by adding additional graphics that capitalize on Mathematica's enhanced three-dimensional graphics capabilities. See especially Example 3.3.15.

- (c) Chapter 4 contains several examples illustrating various techniques for quickly creating plots of bifurcation diagrams, Julia sets, and the Mandelbrot set.
 - (d) The graphics discussion in Chapter 5 has been increased considerably with the addition of Section 5.6, Matrices and Graphs, and the improvement of many of the examples regarding curves and surfaces in space. We have also added a brief discussion regarding the Frenet frame field and curvature and torsion of curves in space. See Examples 5.5.11 and 5.5.12.
 - (e) In Chapter 6, we have taken advantage of the new Manipulate function to illustrate a variety of situations and expand on many examples throughout the chapter. For example, see Example 6.2.5 for a comparison of solutions of nonlinear equations to their corresponding linear approximations.
2. We have included references that we find particularly interesting in the Bibliography, even if they are not specific Mathematica-related texts. A comprehensive list of Mathematica-related publications can be found on the Wolfram website:

<http://store.wolfram.com/catalog/books>

Also, be sure to investigate, use, and support Wolfram's MathWorld, which is simply an amazing web resource for mathematics, Mathematica, and other information.

Finally, we express our appreciation to those who assisted in this project. We express appreciation to our editor, Lauren Schultz, our production editor, Mara Vos-Sarmiento, and our project manager, Phil Bugeau, at Elsevier for providing a pleasant environment in which to work. In addition, Wolfram Research, especially Maryka Baraka, has been most helpful in providing us up-to-date information about Mathematica. Finally, we thank those close to us, especially Imogene Abell, Lori Braselton, Ada Braselton, and Mattie Braselton, for enduring with us the pressures of meeting a deadline and for graciously accepting our demanding work schedules. We certainly could not have completed this task without their care and understanding.

Martha Abell

(email: martha@georgiasouthern.edu)

James Braselton

(email: jbraselton@georgiasouthern.edu)

Statesboro, Georgia

December 2007

This page intentionally left blank

CHAPTER Getting Started

1

1.1 INTRODUCTION TO MATHEMATICA

Mathematica, first released in 1988 by Wolfram Research, Inc. <http://www.wolfram.com/>, is a system for doing mathematics on a computer. Mathematica combines symbolic manipulation, numerical mathematics, outstanding graphics, and a sophisticated programming language. Because of its versatility, Mathematica has established itself as the computer algebra system of choice for many computer users. Among the more than 1 million users of Mathematica, 28% are engineers, 21% are computer scientists, 20% are physical scientists, 12% are mathematical scientists, and 12% are business, social, and life scientists. Two-thirds of the users are in industry and government, and there are a small (8%) but growing number of student users. However, due to its special nature and sophistication, beginning users need to be aware of the special syntax required to make Mathematica perform in the way intended. You will find that calculations and sequences of calculations most frequently used by beginning users are discussed in detail along with many typical examples. In addition, the comprehensive index not only lists a variety of topics but also cross-references commands with frequently used options. *Mathematica by Example* serves as a valuable tool and reference to the beginning user of Mathematica as well as to the more sophisticated user, with specialized needs.

For information, including purchasing information, about Mathematica, contact:

Corporate Headquarters:
Wolfram Research, Inc.
100 Trade Center Drive
Champaign, IL 61820
USA
telephone: 217-398-0700
fax: 217-398-0747
email: info@wolfram.com
website: <http://www.wolfram.com>

Europe:

Wolfram Research Europe Ltd.
10 Blenheim Office Park
Lower Road, Long Hanborough
Oxfordshire OX8 8LN
United Kingdom
telephone: +44-(0) 1993-883400
fax: +44-(0) 1993-883800
email: info-europe@wolfram.com

Asia:

Wolfram Research Asia Ltd.
Izumi Building 8F
3-2-15 Misaki-cho
Chiyoda-ku, Tokyo 101
Japan
telephone: +81-(0)3-5276-0506
fax: +81-(0)3-5276-0509
email: info-asia@wolfram.com

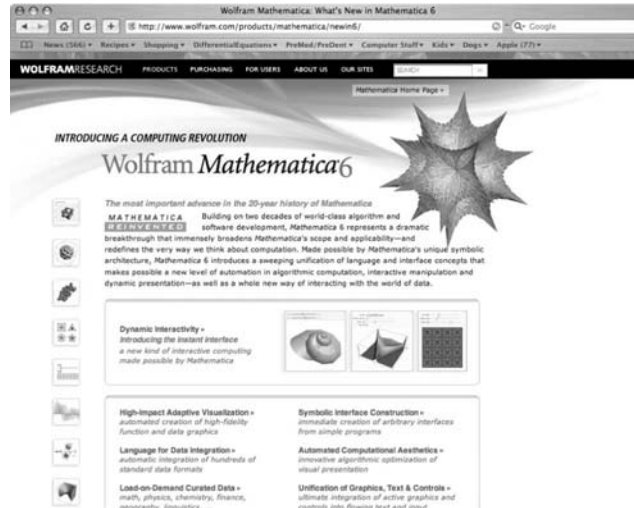
A Note Regarding Different Versions of Mathematica

With the release of Version 6 of Mathematica, many new functions and features have been added to Mathematica. We encourage users of earlier versions of Mathematica to update to Version 6 as soon as possible. All examples in *Mathematica by Example*, fourth edition, were completed with Version 6. In most cases, the same results will be obtained if you are using Version 5.0 or later, although the appearance of your results will almost certainly differ from that presented here. However, particular features of Version 6 are used, and in those cases, of course, these features are not available in earlier versions. If you are using an earlier or later version of Mathematica, your results may not appear in a form identical to those found in this book: Some commands in Version 5 are not available in earlier versions of Mathematica; in later versions, some commands will certainly be changed, new commands added, and obsolete commands removed. For details regarding these changes, please refer to the **Documentation Center**. You can determine the version of Mathematica you are using during a given Mathematica session by entering either the command `$Version` or the command `$VersionNumber`. In this text, we assume that Mathematica has been correctly installed on the computer you are using. If you need to install Mathematica on your computer, please refer to the documentation that came with the Mathematica software package.

On-line help for upgrading older versions of Mathematica and installing new versions is available at the Wolfram Research, website <http://www.wolfram.com/>.

Details regarding what is different in Mathematica 6 from previous versions of Mathematica can be found at

<http://www.wolfram.com/products/mathematica/newin6>



Also, when you go to the **Documentation Center** (under **Help** in the Mathematica menu) you can choose **New in 6** to see the major differences. In addition, the upper right-hand corner of the main help page for each function will tell you if it is new in Version 6 (*New in 6*) or has been updated in Version 6 (*Updated in 6*).

1.1.1 Getting Started with Mathematica

We begin by introducing the essentials of Mathematica. The examples presented are taken from algebra, trigonometry, and calculus topics that you are familiar with to assist you in becoming acquainted with the Mathematica computer algebra system.

We assume that Mathematica has been correctly installed on the computer you are using. If you need to install Mathematica on your computer, please refer to the documentation that came with the Mathematica software package.

Start Mathematica on your computer system. Using Windows or Macintosh mouse or keyboard commands, activate the Mathematica

program by selecting the Mathematica icon or an existing Mathematica document (or notebook) and then clicking or double-clicking on the icon.



If you start Mathematica by selecting the Mathematica icon, a blank untitled notebook is opened, as illustrated in the following screen shot,



along with the **Startup Palette**.



When you start typing, the thin black horizontal line near the top of the window is replaced by what you type.



With some operating systems, **Enter** evaluates commands and **Return** yields a new line.

The **Basic MathInput** palette:



Once Mathematica has been started, computations can be carried out immediately. Mathematica commands are typed and the black horizontal line is replaced by the command, which is then evaluated by pressing **Enter**. Note that pressing **Enter** or **Return** evaluates commands and pressing **Shift-Return** yields a new line. Output is displayed below input. We illustrate some of the typical steps involved in working with Mathematica in the calculations that follow. In each case, we type the command and press **Enter**. Mathematica evaluates the command, displays the result, and inserts a new horizontal line after the result. For example, typing N [, then pressing the π key on the **Basic Math Input** palette, followed by typing, 50] and pressing the enter key

$N[\pi, 50]$

3.1415926535897932384626433832795028841971693993751

returns a 50-digit approximation of π . Note that both π and Pi represent the mathematical constant π , so entering $N[\text{Pi}, 50]$ returns the same result. For basic computations, enter them into Mathematica in the same way as you would with most scientific calculators.

The next calculation can then be typed and entered in the same manner as the first. For example, entering

**$\text{Plot}[\{\text{Sin}[x], 2\text{Cos}[2x]\}, \{x, 0, 3\pi\},$
 $\text{PlotStyle} \rightarrow \{\text{GrayLevel}[0], \text{GrayLevel}[0.5]\}]$**

graphs the functions $y = \sin x$ and $y = 2 \cos 2x$ and on the interval $[0, 3\pi]$ shown in Figure 1.1.

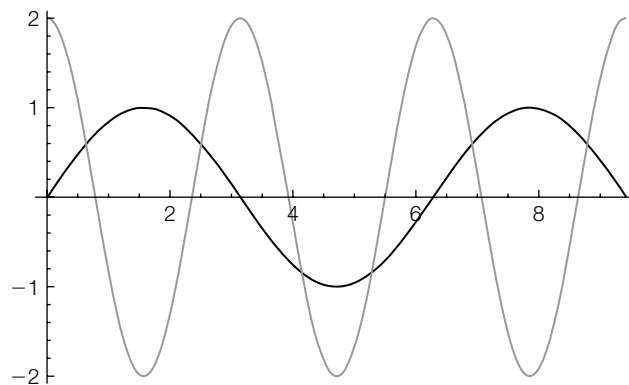


FIGURE 1.1

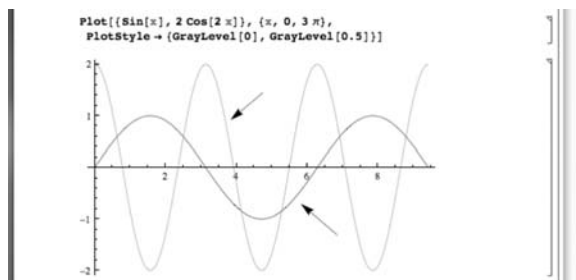
A two-dimensional plot

With Mathematica 6, you can easily add explanation to the graphic. Go to **Graphics** in the main menu, followed by **Drawings Tools**. You can use

the **Drawing Tools** palette to quickly enhance a graphic.



In this case we select the **Arrow** button to add two arrows



and then the **A** button

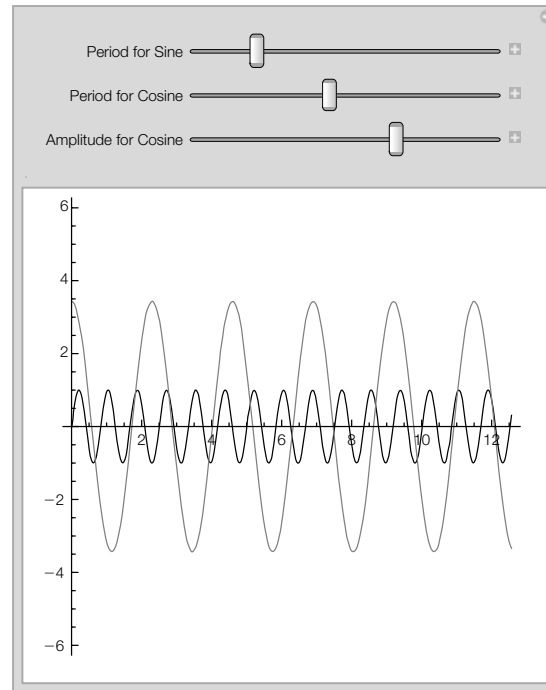
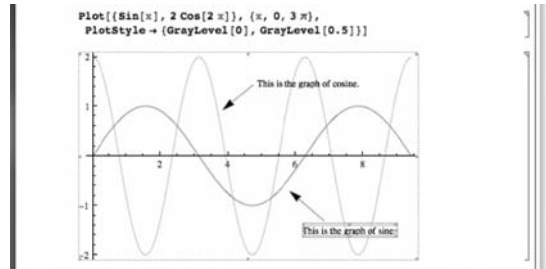


to add some text to help identify each plot. The various elements can be modified by clicking on them and moving and/or typing as needed.

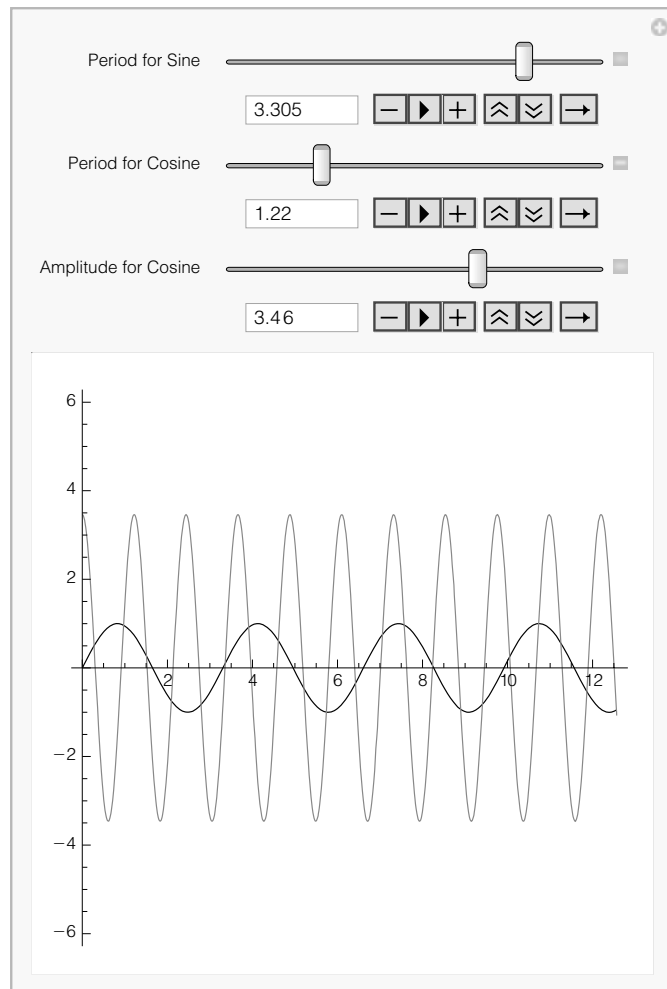
With Mathematica 6, you can use Manipulate to illustrate how changing various parameters affects a given function or functions. With the following

command, we illustrate how a and b affect the period of sine and cosine and c affects the amplitude of cosine:

```
Manipulate[Plot[{Sin[2Pi/ax], cCos[2Pi/bx]}, {x, 0, 4Pi},
  PlotStyle -> {GrayLevel[0], GrayLevel[.5]}, PlotRange -> {-4Pi/2, 4Pi/2},
  AspectRatio -> 1], {{a, 2Pi, "Period for Sine"}, .1, 4},
  {{b, 2Pi, "Period for Cosine"}, .1, 5},
  {{c, 2Pi, "Amplitude for Cosine"}, .1, 5}]
```



Use the slide bars to adjust the values of the parameters or click on the + button to expand the options to enter values explicitly or generate an animation.



Notice that every Mathematica command begins with capital letters and the argument is enclosed by square brackets [...].

Use Plot3D to generate basic three-dimensional plots. Entering

Plot3D[Sin[x + Cos[y]], {x, 0, 4π}, {y, 0, 4π}, Ticks → None, Boxed → False, Axes → None]

graphs the function $z = \sin(x + \cos y)$ for $0 \leq x \leq 4\pi$ and $0 \leq y \leq 4\pi$ shown in Figure 1.2. To view the image from different angles, use the mouse to select the graphic and then drag to the desired angle.

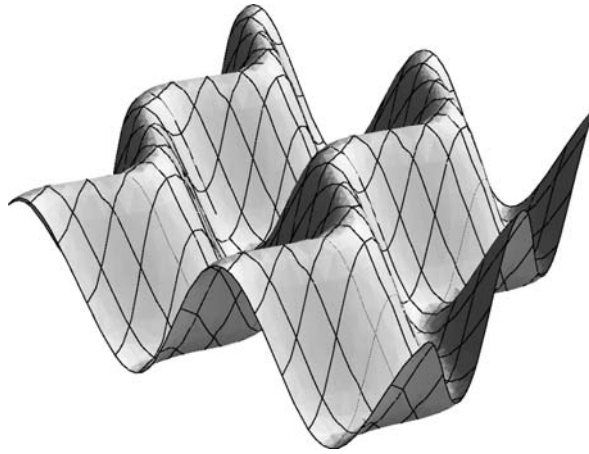
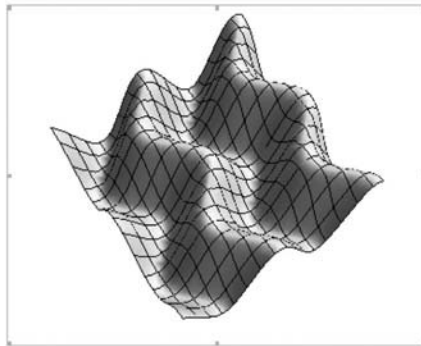


FIGURE 1.2

A three-dimensional plot

To type x^3 in Mathematica, press the  on the **Basic Math Input** palette, type x in the base position, and then click (or tab to) the exponent position and type 3. Use the **esc** key, tab button, or mouse to help you place or remove the cursor from its current location.

```
Plot3D[Sin[x + Cos[y]], {x, 0, 4 π}, {y, 0, 4 π}, Ticks → None,
Boxed → False, Axes → None]
```



Notice that all three of the following commands

```
Solve[x^3 - 2 x + 1 == 0]
{{x -> 1}, {x -> 1/2 (-1 - Sqrt[5])}, {x -> 1/2 (-1 + Sqrt[5])}}
```

```
Solve[x^3 - 2 x + 1 == 0]
{{x -> 1}, {x -> 1/2 (-1 - Sqrt[5])}, {x -> 1/2 (-1 + Sqrt[5])}}
```

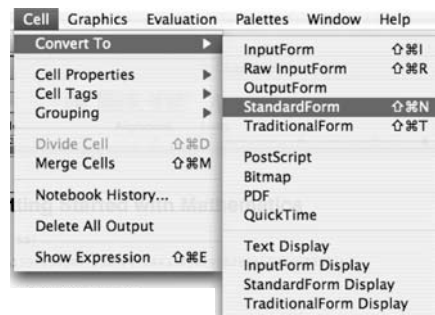
```
Solve[x^3 - 2 x + 1 == 0]
{{x -> 1}, {x -> 1/2 (-1 - Sqrt[5])}, {x -> 1/2 (-1 + Sqrt[5])}}
```

solve the equation $x^3 - 3x + 1 = 0$ for x .

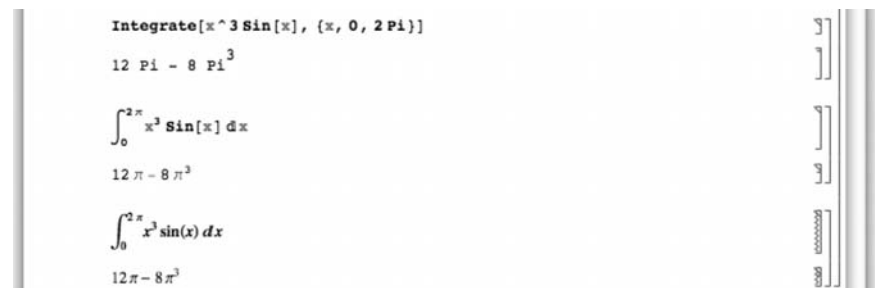
In the first case, the input and output are in **StandardForm**; in the second case, the input and output are in **InputForm**; and in the third case, the input and output are in **TraditionalForm**. Move the cursor to the Mathematica menu,



select **Cell**, and then **ConvertTo**, as illustrated in the following screen shot:



You can change how input and output appear by using **ConvertTo** or by changing the default settings. Moreover, you can determine the form of input/output by looking at the cell bracket that contains the input/output. For example, even though all three of the following commands look different, all three evaluate $\int_0^{2\pi} x^3 \sin x \, dx$:



In the first calculation, the input is in **InputForm** and the output in **OutputForm**; in the second, the input and output are in **StandardForm**; and in the third, the input and output are in **TraditionalForm**. Throughout *Mathematica by Example*, fourth edition, we display input and output using **InputForm** (for input) or **StandardForm** (for output), unless otherwise stated.

To enter code in **StandardForm**, we often take advantage of the **Basic Math Input** palette, which is accessed by going to **Palettes** under the Mathematica menu and then selecting **BasicMathInput**. See Figure 1.3.

Use the buttons to create templates and enter special characters. Alternatively, you can access a complete list of typesetting shortcuts from Mathematica help at `guide/MathematicalTypesetting` in the **Documentation Center**.

Mathematica sessions are terminated by entering `Quit[]` or by selecting **Quit** from the **File** menu, or by using a keyboard shortcut, such as **command-Q**, as with other applications. They can be saved by referring to **Save** from the **File** menu.

Mathematica allows you to save notebooks (as well as combinations of cells) in a variety of formats, in addition to the standard Mathematica format.

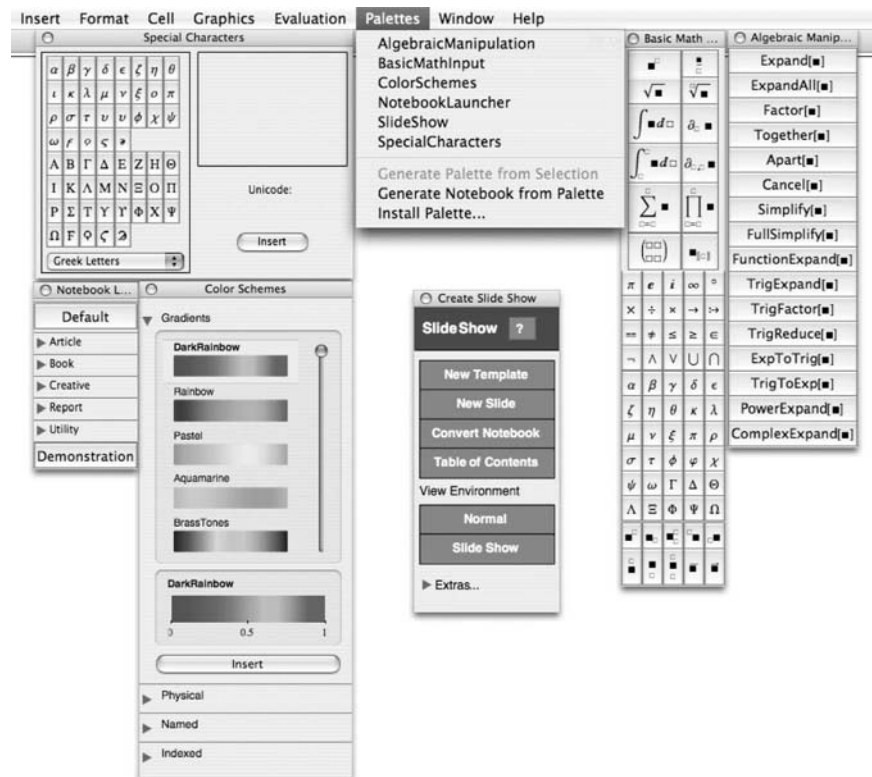


FIGURE 1.3

Mathematica 6 palettes



Remark 1.1 Input and text regions in notebooks can be edited. Editing input can create a notebook in which the mathematical output does not make sense in the sequence it appears. It is also possible to simply go into a notebook and alter input without doing any recalculation. This also creates misleading notebooks. Hence, common sense and caution should be used when editing the input regions of notebooks. Recalculating all commands in the notebook will clarify any confusion.

Preview

In order for the Mathematica user to take full advantage of this powerful software, an understanding of its syntax is imperative. The goal of *Mathematica by Example* is to introduce the reader to the Mathematica commands and sequences of commands most frequently used by beginning users. Although the rules of Mathematica syntax are far too numerous to list here, knowledge of the following five rules equips the beginner with the necessary tools to start using the Mathematica program with little trouble.

Five Basic Rules of Mathematica Syntax

1. The arguments of *all* functions (both built-in ones and ones that you define) are given in brackets [. . .]. Parentheses (. . .) are used for grouping operations; vectors, matrices, and lists are given in braces { . . . }; and double square brackets [[. . .]] are used for indexing lists and tables.
2. Every word of a built-in Mathematica function begins with a capital letter.
3. Multiplication is represented by a * or space between characters. Enter $2*x*y$ or $2x\ y$ to evaluate $2xy$ *not* $2xy$.
4. Powers are denoted by a ^ . Enter $(8*x^3)^{(1/3)}$ to evaluate $(8x^3)^{1/3} = 8^{1/3}(x^3)^{1/3} = 2x$ instead of $8x^{1/3}$, which returns $8x/3$.
5. Mathematica follows the order of operations *exactly*. Thus, entering $(1 + x)^{1/x}$ returns $\frac{(1+x)^1}{x}$, whereas $(1 + x)^{(1/x)}$ returns $(1 + x)^{1/x}$. Similarly, entering x^3x returns $x^3 \cdot x = x^4$, whereas entering $x^{(3x)}$ returns x^{3x} .

Remark 1.2 If you get no response or an incorrect response, you may have entered or executed the command incorrectly. In some cases, the amount of memory allocated to Mathematica can cause a crash. Like people, Mathematica is not perfect and errors can occur.

1.2 LOADING PACKAGES

Although Mathematica contains many built-in functions, some other functions are contained in **packages** that must be loaded separately. Experienced users can create their own packages; other packages are available from user groups and **MathSource**, which electronically distributes Mathematica-related products. For information about MathSource, visit

<http://library.wolfram.com/infocenter/MathSource>

or send the message “help” to mathsource@wri.com. If desired, you can purchase MathSource on a CD directly from Wolfram Research, or you can access MathSource from the Wolfram Research website.

With Mathematica 6, many packages included with previous versions of Mathematica have been made obsolete because their functionality has been incorporated into Mathematica, combined into a new package, or eliminated altogether. In addition to **MathSource**, you should also think about investigating Wolfram’s **MathWorld** website.

1.2.1 Packages Included with Older Versions of Mathematica

Packages are loaded by entering the command `<<directory`packagename`, Needs[directory`packagename], <<packagename` or Needs[packagename]`, where **directory** is the location of the package **packagename**. Entering the command `<<directory`Master`` makes all the functions contained in all the packages in **directory** available. In this case, each package need not be loaded individually.

Over time, Wolfram Research expects that packages will become obsolete and that Mathematica will automatically load functions that are needed at startup or when called.

For most teachers and students, a function like $f(x) = (x - 1)^{1/3}(x + 1)^{2/3}$ is a real-valued function for all values of x . Nevertheless, when we ask Mathematica to plot the function with `Plot`,

```
Plot[(x - 1)^(1/3)(x + 1)^(2/3), {x, -2, 2}, PlotStyle -> GrayLevel[0]]
```

we see in Figure 1.4 that Mathematica does not compute real values for x values between -1 and 1 because complex roots are selected by Mathematica for the x values between -1 and 1 , which is where the values of $f(x)$ are negative.

Generally, when Mathematica computes the odd root of a negative number, it returns a complex number. (Note that `%` refers to the previous

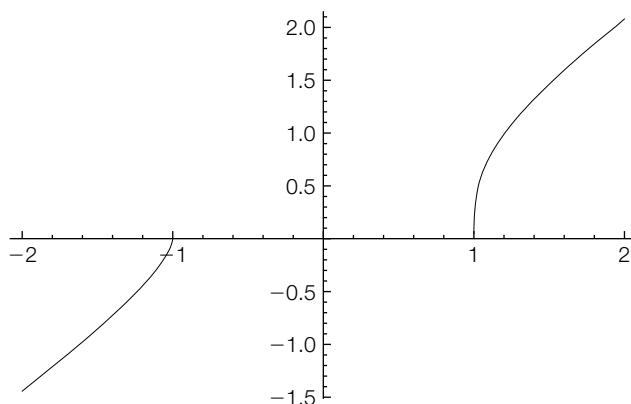


FIGURE 1.4

When computing odd roots of negative numbers, Mathematica returns complex values

output; `N[x]` returns a numerical approximation of x , and `Abs[x]` returns the absolute value of the number x .)

```
(-8)^(1/3)
```

```
2(-1)^(1/3)
```

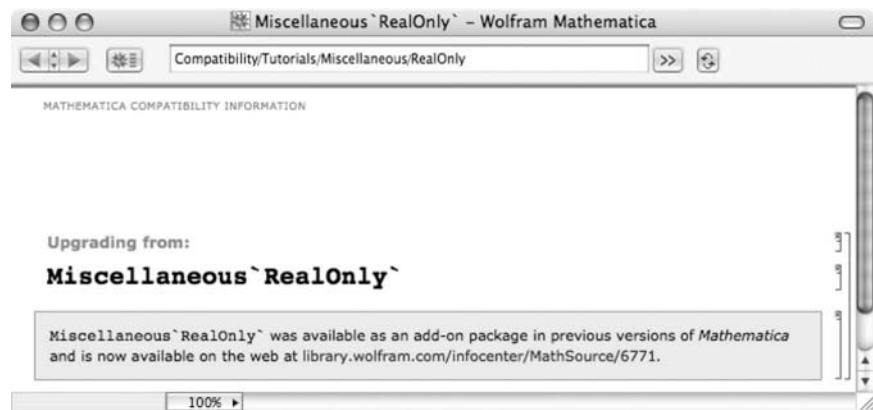
```
N[%]
```

```
1. + 1.73205i
```

```
Abs[%]
```

```
2.
```

To instruct Mathematica to select the *real* third root, we load the **RealOnly** package that is contained in the **Miscellaneous** directory. Note that the **RealOnly** package has been included with many versions of Mathematica but *not* included with Mathematica 6. If you need to obtain the **RealOnly** package, you need to download it from the Wolfram website.



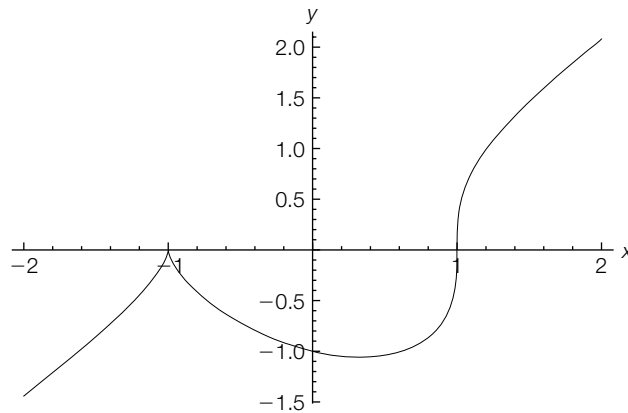
After loading the package, when we reenter the `Plot` command, Mathematica generates the expected plot, which is shown in Figure 1.5.

```
<< Miscellaneous`RealOnly`
```

```
Plot[(x - 1)^(1/3)(x + 1)^(2/3), {x, -2, 2}, PlotStyle -> GrayLevel[0]]
```

1.2.2 Loading New Packages

One new package included with Mathematica 6 is **VectorFieldPlots**, which replaces several packages in previous versions of Mathematica.

**FIGURE 1.5**

We see the real values of $f(x)$ for $-1 < x < 1$ after loading the **RealOnly** package



Example 1.2.1 The differential equation $dy/dx = \cos(y/x)$ is a **first-order homogeneous differential equation**. Using DSolve, we see that the solution contains an integral that does not have a known closed form. The result returned by DSolve indicates

that the integral curves for the differential equation satisfy the equation contained within the brackets in the output:

DSolve[y'[x] == Cos[y[x]/x], y[x], x]

Solve::tdep: The equations appear to involve the variables to be solved for in an essentially nonalgebraic way. >>

$$\text{Solve} \left[\int_1^{\frac{y[x]}{x}} \frac{1}{-\text{Cos}[K[1]] + K[1]} dK[1] == C[1] - \text{Log}[x], y[x] \right]$$

For a differential equation like this, even the function $g(x, y) = \int_1^{y/x} \frac{1}{t - \cos t} dt + \ln|x|$ is difficult to evaluate for particular values of x and y , so generating a plot of the level curves of $g(x, y) = C$ (the integral curves for the differential equation) for various values of C is challenging.

To see how the solutions of the differential equation behave, we plot a **direction field** or **slope field** for the equation. For this equation, the slope of a solution at (x, y) satisfies $dy/dx = \cos(y/x)$. A direction field for the equation is generated by selecting a grid of (x, y) points and then plotting line segments at those points with slope $dy/dx = \cos(y/x)$. With Mathematica, we can do so with the **VectorFieldPlot** function that is contained in the **VectorFieldPlots** package. First, we load the package with

<< VectorFieldPlots`;

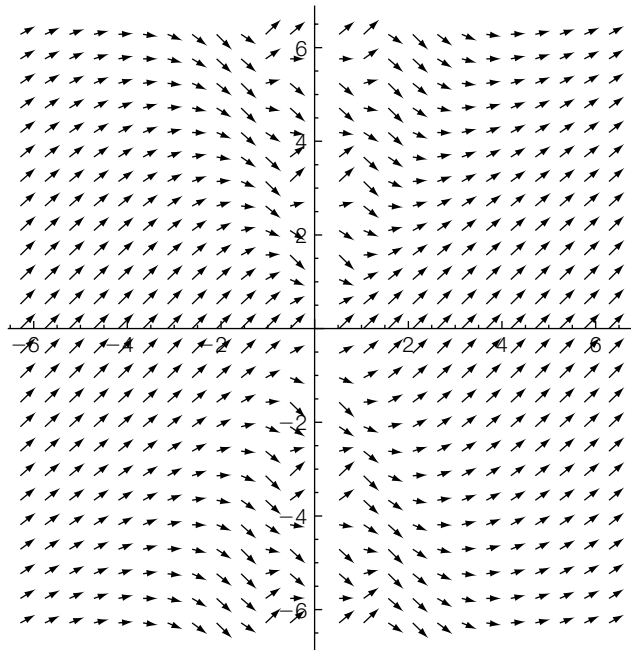
Now that the package has been loaded, you can use **?** or **Options** to obtain information about the commands contained in the package. Finally, we generate a slope field for the equation with

**p1 = VectorFieldPlot[{1, f[x, y]}, {x, -2Pi, 2Pi}, {y, -2Pi, 2Pi},
PlotPoints -> 25];
Show[p1, Axes -> Automatic, AxesOrigin -> {0, 0}]**

Note that Mathematica returns several error messages due to the division by 0 in the y/x term that are not displayed here. The plot is displayed in Figure 1.6. From the slope field, we see that solutions of the differential equation can behave quite strangely near $x = 0$.

1.3 GETTING HELP FROM MATHEMATICA

Becoming competent with Mathematica can take a serious investment of time. Hopefully, messages that result from syntax errors are viewed light-heartedly. Ideally, instead of becoming frustrated, beginning Mathematica users will find it challenging and fun to locate the source of errors. Frequently, Mathematica's error messages indicate where the error(s) has occurred. In this process, it is natural that you will become more proficient

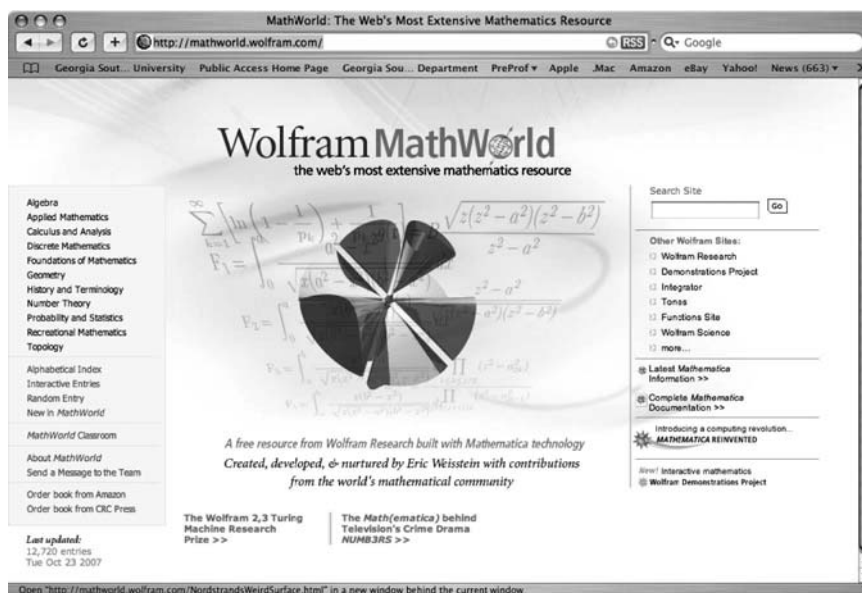
**FIGURE 1.6**

Numerically solving a differential equation such as $dy/dc = \cos(y/x)$ is difficult. To help us understand how the solutions behave, we use a slope field

with Mathematica. In addition to Mathematica's extensive help facilities, which are described next, a tremendous amount of information is available for all Mathematica users at the Wolfram Research website. Not only can you get significant Mathematica help at the Wolfram website but also you can access outstanding *mathematical* resources at Wolfram's MathWorld resource,

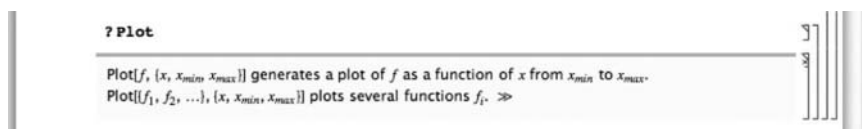
<http://mathworld.wolfram.com>

One way to obtain information about Mathematica commands and functions, including user-defined functions, is the command `?object` gives a basic description and syntax information of the Mathematica object `object`. `??object` yields detailed information regarding syntax and options for the object `object`. Equivalently, `Information[object]` yields the information on the Mathematica object `object` returned by both `?object` and `Options[object]` in addition to a list of attributes of `object`. Note that `object` may either be a user-defined object or a built-in Mathematica object.



Example 1.3.1 Use `?` and `??` to obtain information about the command `Plot`.

Solution `?Plot` uses basic information about the `Plot` function,



whereas `??Plot` includes basic information as well as a list of options and their default values.

If you click on the `>>` button, Mathematica returns its extensive description of the function. Notice that the updated button in Version 6 (*Updated in 6*) shows that `Plot` has been updated. Click on **Show Changes** and then **More Information** to see the changes in Version 6.

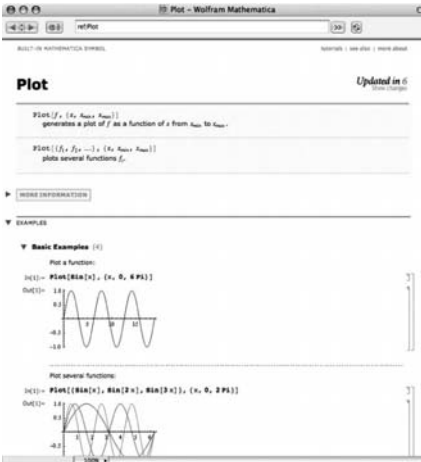
`Options[object]` returns a list of the available options associated with objects along with their current settings. This is quite useful when working with a Mathematica command such as `ParametricPlot`, which has many options. Notice that the default value (the value automatically assumed by Mathematica) for each option is given in the output.

?? Plot

`Plot[f, {x, xmin, xmax}]` generates a plot of f as a function of x from $x_{\min}$ to $x_{\max}$.
`Plot[{f1, f2, ...}, {x, xmin, xmax}]` plots several functions f_i . >>

`Attributes[Plot] = {HoldAll, Protected}`

`Options[Plot] = {AlignmentPoint → Center, AspectRatio → $\frac{1}{\text{GoldenRatio}}$,
 Axes → True, AxesLabel → None, AxesOrigin → Automatic,
 AxesStyle → {}, Background → None, BaselinePosition → Automatic,
 BaseStyle → {}, ClippingStyle → None, ColorFunction → Automatic,
 ColorFunctionScaling → True, ColorOutput → Automatic,
 ContentSelectable → Automatic, DisplayFunction → $DisplayFunction,
 Epilog → {}, Evaluated → System`Private`Evaluated,
 EvaluationMonitor → None, Exclusions → Automatic,
 ExclusionsStyle → None, Filling → None, FillingStyle → Automatic,
 FormatType → TraditionalForm, Frame → False, FrameLabel → None,
 FrameStyle → {}, FrameTicks → Automatic, FrameTicksStyle → {},
 GridLines → None, GridLinesStyle → {}, ImageMargins → 0.,
 ImagePadding → All, ImageSize → Automatic, LabelStyle → {},
 MaxRecursion → Automatic, Mesh → None, MeshFunctions → {1 &},
 MeshShading → None, MeshStyle → Automatic, Method → Automatic,
 PerformanceGoal → $PerformanceGoal, PlotLabel → None,
 PlotPoints → Automatic, PlotRange → {Full, Automatic},
 PlotRangeClipping → True, PlotRangePadding → Automatic,
 PlotRegion → Automatic, PlotStyle → Automatic,
 PreserveImageOptions → Automatic, Prolog → {},
 RegionFunction → {True &}, RotateLabel → True, Ticks → Automatic,
 TicksStyle → {}, WorkingPrecision → MachinePrecision}`





Example 1.3.2 Use `Options` to obtain a list of the options and their current settings for the command `ParametricPlot`.

Solution The command `Options [ParametricPlot]` lists all the options and their current settings for the command `ParametricPlot`.

```
Options[ParametricPlot]
{AlignmentPoint -> Center, AspectRatio -> Automatic, Axes -> True,
 AxesLabel -> None, AxesOrigin -> Automatic, AxesStyle -> {},
 Background -> None, BaselinePosition -> Automatic, BaseStyle -> {},
 BoundaryStyle -> Automatic, ColorFunction -> Automatic,
 ColorFunctionScaling -> True, ColorOutput -> Automatic,
 ContentSelectable -> Automatic, DisplayFunction -> $DisplayFunction,
 Epilog -> {}, Evaluated -> Automatic, EvaluationMonitor -> None,
 Exclusions -> Automatic, ExclusionsStyle -> None,
 FormatType -> TraditionalForm, Frame -> Automatic, FrameLabel -> None,
 FrameStyle -> {}, FrameTicks -> Automatic, FrameTicksStyle -> {},
 GridLines -> None, GridLinesStyle -> {}, ImageMargins -> 0.,
 ImagePadding -> All, ImageSize -> Automatic, LabelStyle -> {},
 MaxRecursion -> Automatic, Mesh -> Automatic, MeshFunctions -> Automatic,
 MeshShading -> None, MeshStyle -> Automatic, Method -> Automatic,
 PerformanceGoal -> $PerformanceGoal, PlotLabel -> None,
 PlotPoints -> Automatic, PlotRange -> Automatic, PlotRangeClipping -> True,
 PlotRangePadding -> Automatic, PlotRegion -> Automatic,
 PlotStyle -> Automatic, PreserveImageOptions -> Automatic, Prolog -> {},
 RegionFunction -> (True &), RotateLabel -> True, Ticks -> Automatic,
 TicksStyle -> {}, WorkingPrecision -> MachinePrecision}
```

The command `Names["form"]` lists all objects that match the pattern defined in `form`. For example, `Names["Plot"]` returns `Plot`, `Names["*Plot"]` returns all objects that end with the string `Plot`, `Names["Plot*"]` lists all objects that begin with the string `Plot`, and `Names["*Plot*"]` lists all objects that contain the string `Plot`. `Names["form", SpellingCorrection -> True]` finds those symbols that match the pattern defined in `form` after a spelling correction.

Example 1.3.3 Create a list of all built-in functions beginning with the string `Plot`.

Solution We use `Names` to find all objects that match the pattern `Plot`.

```
Names["Plot"]
{Plot}
```

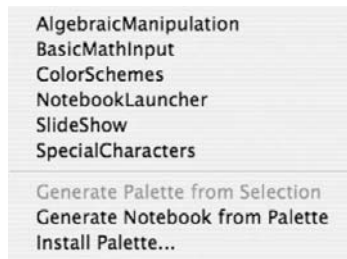
Next, we use `Names` to create a list of all built-in functions beginning with the string `Plot`.

```
Names["Plot*"]
{Plot, Plot3D, Plot3Matrix, PlotDivision,
 PlotJoined, PlotLabel, PlotMarkers, PlotPoints, PlotRange,
 PlotRangeClipping, PlotRangePadding, PlotRegion, PlotStyle}
```

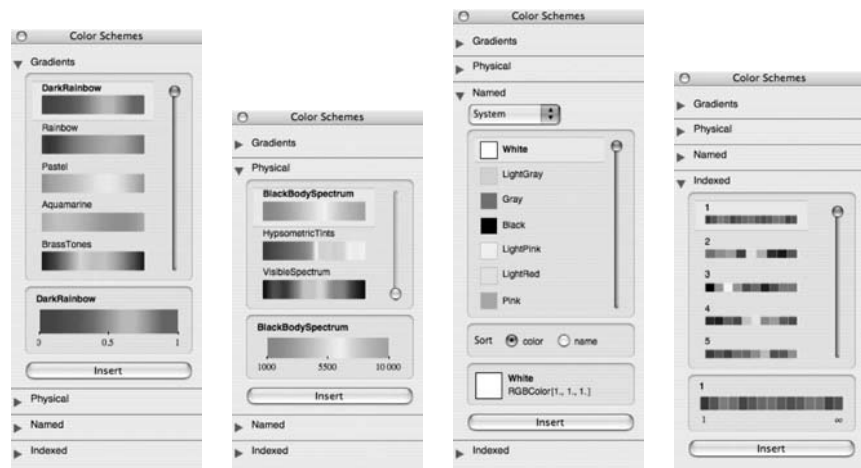
In the following, after using ? to learn about the new Mathematica 6 function `ColorData` we illustrate its use with a `Plot` command. We first go to the Mathematica menu



and select **Palettes**, followed by **ColorSchemes**.



Remember that on a computer running Mathematica, these graphics will appear in color rather than in black-and-white as seen in this text.

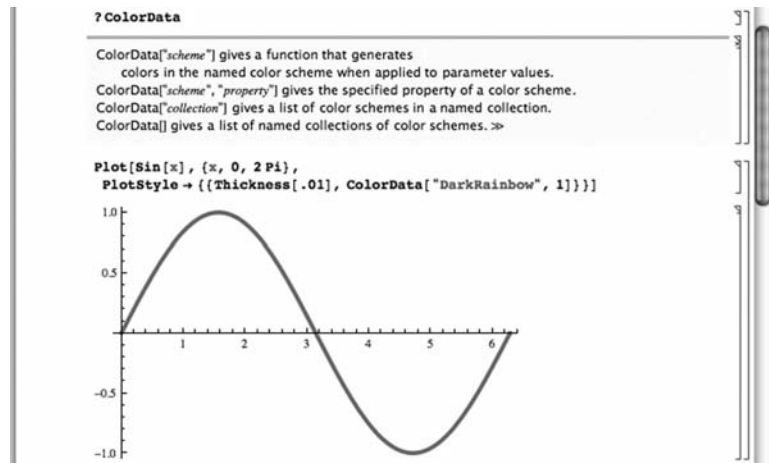


We then use the help facilities description of the **ColorData** function to help us generate a plot of $y = \sin x$ on the interval $[0, 2\pi]$ in deep red on our computer. (Of course, the plot is dark gray in a black-and-white text such as this).

As we have illustrated, the `?function` can be used in many ways. Entering `?letters*` gives all Mathematica objects that begin with the string `letters`; `?*letters*` gives all Mathematica objects that contain the string `letters`; and `?*letters` gives all Mathematica commands that end in the string `letters`.

Example 1.3.4 What are the Mathematica functions that (a) end in the string `Cos`, (b) contain the string `Sin`, and (c) begin with the string `Polynomial`?

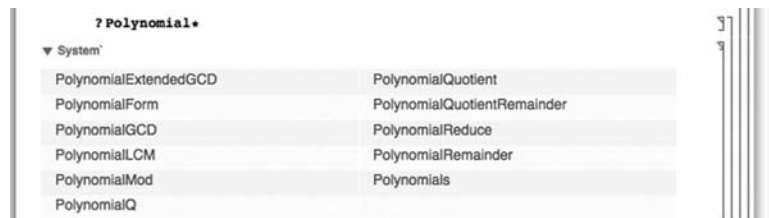
Solution Entering



returns all functions ending with the string Cos, entering



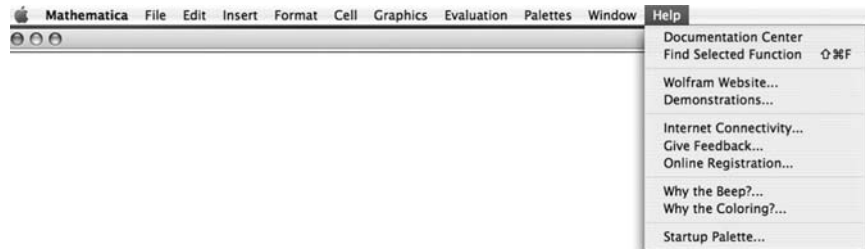
returns all functions containing the string Sin, and entering



returns all functions that begin with the string Polynomial.

Mathematica Help

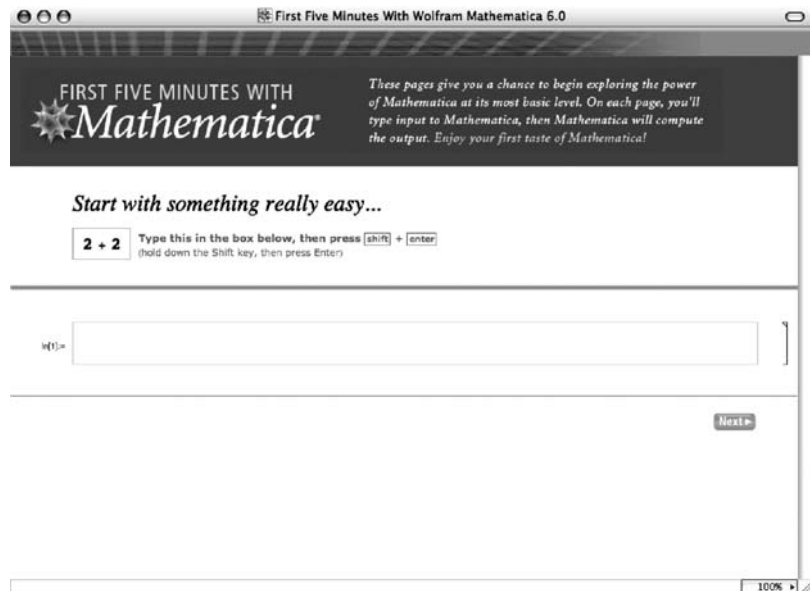
Additional help features are accessed from the Mathematica menu under **Help**. For basic information about Mathematica, go to the Mathematica menu, followed by **Help**



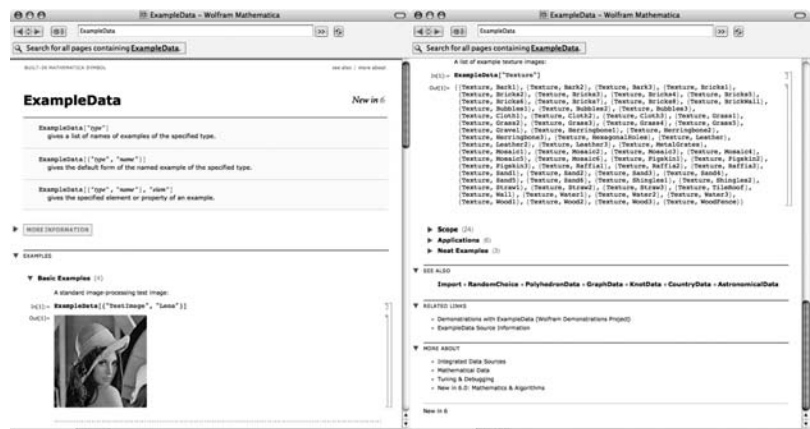
and select **Documentation Center**.



If you are a beginning Mathematica user, you may choose to select **First Five Minutes with Mathematica**.



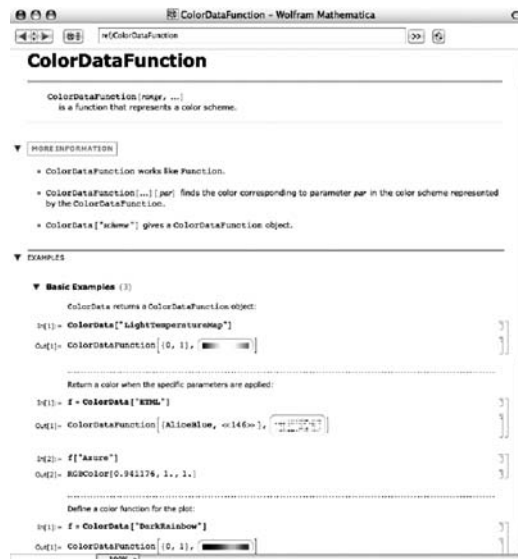
To obtain information about a particular Mathematica object or function, open the **Documentation Center**, type the name of the object, function, or topic, and press the **Go (>>)** button as we have done here with ExampleData. A typical help window contains not only a detailed description of the command and its options but also hyperlinked cross-references to related commands and can be accessed by clicking on the appropriate links.



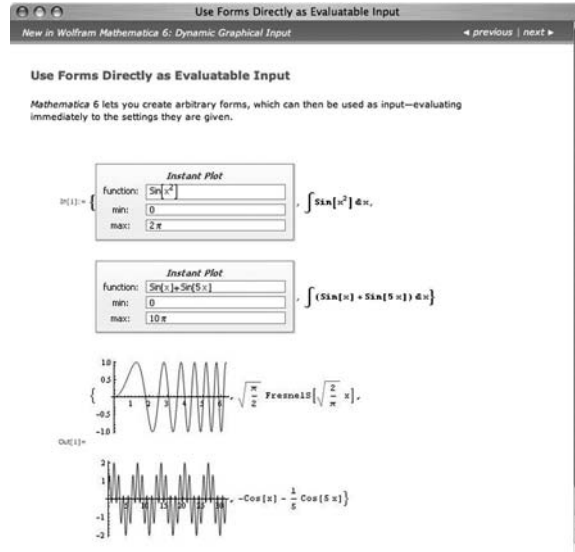
You can also use the **Documentation Center** to search for help regarding a particular topic. In this case, we enter color schemes in the top line of the **Documentation Center** and then click on the >> button (or press **Enter**) to see all the on-line help regarding “color schemes.”



Clicking on the topic will take you to the documentation for the topic. Here is what we see when we select **ColorDataFunction**:



As you become more proficient with Mathematica, you will want to learn to take advantage of its extensive capabilities.



Remember that Mathematica contains thousands of functions to perform many tasks. If you wish to perform a task that is not discussed here, go to the **Documentation Center** and type a few words related to what you want to do.

Example 1.3.5 In this example, we investigate digit operations. *Mathematica by Example*, fourth edition, has a copyright in 2008, which has four digits.

IntegerDigits[2008]

{2,0,0,8}

As a string, the number is

IntegerString[2008]

2008

In base 2, the copyright year is

IntegerString[2008, 2]

11111011000

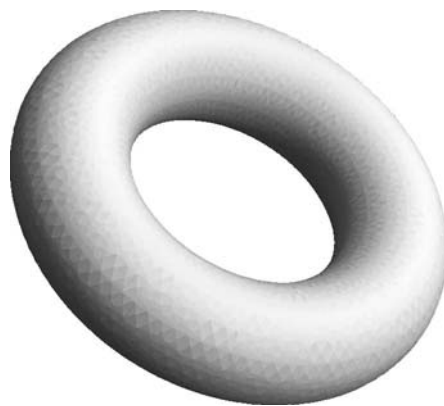
On the other hand, with Roman numerals the copyright year is

IntegerString[2008, "Roman"]

MMVIII

1.4 EXERCISES

1. Generate the same plot of $f(x) = (x - 1)^{1/3}(x + 1)^{2/3}$ as that shown *after* loading the **RealOnly** package *without* loading the **RealOnly** package. Hint: `Abs[x]` returns the absolute value of the real number x .
2. After using the **Documentation Center** to obtain help regarding the function `Plot`, describe the use of the `Plot` function.
3. Use Mathematica help and the **Documentation Center** to describe the `ExampleData` function. Use `ExampleData` to generate the plot of the torus shown in Figure 1.7.
4. Use `?` to determine the value of the Golden Ratio, `GoldenRatio`.
5. Determine the proper syntax for evaluating $\lim_{x \rightarrow \pi/2} \sin x$ and evaluate the limit.
6. Load the **VectorFieldPlot** package. Use `Options` to learn about the options associated with `VectorFieldPlot`. Describe the use of three of those options. Your description should contain sufficient detail so that it is readily understandable by an intelligent classmate.
7. Find the graphing options available with `Plot3D` and `ParametricPlot3D`.
8. Determine the Mathematica objects that contain the string “gamma.”
9. Do any Mathematica objects begin with the letter “z”? Do any end with “z”?
10. What Mathematica function is used to represent the inverse tangent function?
11. Create a list of *all* Mathematica objects.

**FIGURE 1.7**

A plot of a torus generated with `ExampleData`

12. Visit MathWorld at <http://mathworld.wolfram.com>. Use RandomInteger to generate a random integer n between 1 and 11. Visit the n th mathematical topic in the subject list. Then, randomly visit a subtopic followed by another subtopic. From the list of topics, choose one that sounds interesting but that you know nothing about. Follow the link and learn about the topic. Write a brief (one-page) report on your findings.

This page intentionally left blank

Basic Operations on Numbers, Expressions, and Functions

Chapter 2 introduces the essential commands of Mathematica. Basic operations on numbers, expressions, and functions are introduced and discussed.

2.1 NUMERICAL CALCULATIONS AND BUILT-IN FUNCTIONS

2.1.1 Numerical Calculations

The basic arithmetic operations (addition, subtraction, multiplication, division, and exponentiation) are performed in the natural way with Mathematica. Whenever possible, Mathematica gives an exact answer and reduces fractions.

1. “ a plus b ,” $a + b$, is entered as $a + b$;
2. “ a minus b ,” $a - b$, is entered as $a - b$;
3. “ a times b ,” ab , is entered as either $a * b$ or $a \ b$ (note the space between the symbols a and b);
4. “ a divided by b ,” a/b , is entered as a/b . Executing the command a/b results in a fraction reduced to lowest terms; and
5. “ a raised to the b th power,” a^b , is entered as a^b .

Example 2.1.1 Calculate (a) $121 + 542$; (b) $3231 - 9876$; (c) $(-23)(76)$; (d) $(22341)(832748)(387281)$; and (e) $\frac{467}{31}$.

Solution These calculations are carried out in the following screen shot. In each case, the input is typed and then evaluated by pressing **Enter**. In the last case, the **Basic Math** template is used to enter the fraction.

Basic Math ...

$\frac{\square}{\square}$	$\frac{\square}{\square}$
$\sqrt{\square}$	$\sqrt[n]{\square}$
$\int \square d\square$	$\partial \square$
$\int \square d\square$	$\partial \square$
$\sum \square$	$\prod \square$
$(\square)$	$\square[\square]$

π	e	i	∞	$\circ$
$\times$	$\div$	$\times$	$\rightarrow$	$\rightarrow$
$=$	$\neq$	$\leq$	$\geq$	$\in$
$\neg$	$\wedge$	$\vee$	$\cup$	$\cap$
α	β	γ	δ	ϵ
ζ	η	θ	κ	λ
μ	ν	ξ	π	ρ
σ	τ	ϕ	φ	χ
ψ	ω	Γ	Δ	Θ
Λ	Ξ	Φ	Ψ	Ω
$\square$	$\square$	$\square$	$\square$	$\square$
$\square$	$\square$	$\square$	$\square$	$\square$

■ **Example 1**

121 + 452
573

3231 - 9876
-6645

-23 * 76
-1748

22361 * 832748 * 387281
7211589719761868

467 / 31
467
31

467
31
467
31

The term $a^{n/m} = \sqrt[m]{a^n} = (\sqrt[n]{a})^m$ is entered as $a^{(n/m)}$. For $n/m = 1/2$, the command $\text{Sqrt}[a]$ can be used instead. Usually, the result is returned in unevaluated form but N can be used to obtain numerical approximations to virtually any degree of accuracy. With $N[\text{expr}, n]$, Mathematica yields a numerical approximation of expr to n digits of precision, if possible. At other times, Simplify can be used to produce the expected results.

Remark 2.1 If the expression b in a^b contains more than one symbol, be sure that the exponent is included in parentheses. Entering $a^{n/m}$ computes $a^{n/m} = \frac{1}{m}a^n$, whereas entering $a^{(n/m)}$ computes $a^{n/m}$.

Example 2.1.2 Compute (a) $\sqrt{27}$ and (b) $\sqrt[3]{8^2} = 8^{2/3}$.

Solution (a) Mathematica automatically simplifies $\sqrt{27} = 3\sqrt{3}$. We use N to obtain an approximation of $\sqrt{27}$. (b) Mathematica automatically simplifies $8^{2/3}$.

`N[number]` and `number//N` return numerical approximations of number.

```
Sqrt[27]
3 Sqrt[3]

N[Sqrt[27]]
5.19615

8^(2/3)
4
```

```
]]
]]
]]
]]
]]
```

When computing odd roots of negative numbers, Mathematica's results are surprising to the novice. Namely, Mathematica returns a complex number. We will see that this has important consequences when graphing certain functions.

Example 2.1.3 Calculate (a) $\frac{1}{3} \left(-\frac{27}{64}\right)^2$ and (b) $\left(-\frac{27}{64}\right)^{2/3}$.

Solution (a) Because Mathematica follows the order of operations, $(-27/64)^{2/3}$ first computes $(-27/64)^2$ and then divides the result by 3.

```
(-27 / 64) ^ 2 / 3
243
4096
```

```
]]
]]
```

(b) On the other hand, $(-27/64)^{(2/3)}$ raises $-27/64$ to the $2/3$ power. Mathematica does not automatically simplify $(-27/64)^{2/3}$.

```
(-27 / 64) ^ (2 / 3)
9
16 (-1)^(2/3)
```

```
]]
]]
```

However, when we use `N`, Mathematica returns the numerical version of the principal root of $(-27/64)^{2/3}$.

```
N[(-27 / 64) ^ (2 / 3)]
-0.28125 + 0.487139 I
```

```
]]
]]
```

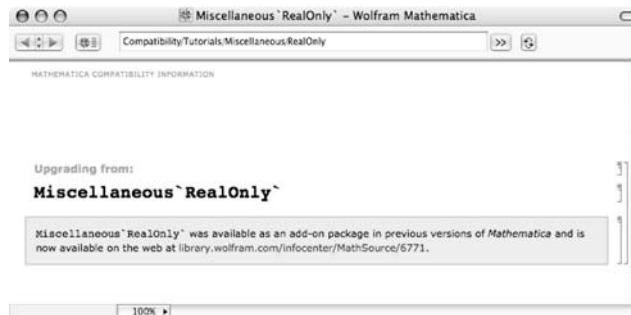
To obtain the result

$$\left(-\frac{27}{64}\right)^{2/3} = \left(\sqrt[3]{-\frac{27}{64}}\right)^2 = \left(-\frac{3}{4}\right)^2 = \frac{9}{16},$$

which would be expected by most algebra and calculus students, we first square $-27/64$ and then take the third root.

$$\frac{((-27/64)^2)^{1/3}}{16}$$

Alternatively, download the **RealOnly** package from the Wolfram website.



Then,

```
<< Miscellaneous`RealOnly`
General::obspkg:
  Miscellaneous`RealOnly` is now obsolete. The legacy version being loaded
  may conflict with current Mathematica functionality.
  See the Compatibility Guide for updating information. >>

((-27 / 64) ^ (2 / 3))
9
16
```

returns the result 9/16.

2.1.2 Built-in Constants

Mathematica has built-in definitions of nearly all commonly used mathematical constants and functions. To list a few, $e \approx 2.71828$ is denoted by `E`, $\pi \approx 3.14159$ is denoted by `Pi`, and $i = \sqrt{-1}$ is denoted by `I`. Usually, Mathematica performs complex arithmetic automatically.

Other built-in constants include ∞ ; denoted by `Infinity`; Euler's constant, $\gamma \approx 0.577216$, denoted by `EulerGamma`; Catalan's constant, approximately 0.915966, denoted by `Catalan`; and the golden ratio, $\frac{1}{2}(1 + \sqrt{5}) \approx 1.61803$, denoted by `GoldenRatio`.

Example 2.1.4 Entering**N[E, 50]**

2.7182818284590452353602874713526624977572470937000

returns a 50-digit approximation of e . Entering**N[π , 25]**

3.141592653589793238462643

returns a 25-digit approximation of π . Entering

$$\frac{3+i}{4-i}$$

$$\frac{11}{17} + \frac{7i}{17}$$

performs the division $(3+i)/(4-i)$ and writes the result in standard form.**2.1.3 Built-in Functions**

Functions frequently encountered by beginning users include the exponential function, $\text{Exp}[x]$; the natural logarithm, $\text{Log}[x]$; the absolute value function, $\text{Abs}[x]$; the trigonometric functions $\text{Sin}[x]$, $\text{Cos}[x]$, $\text{Tan}[x]$, $\text{Sec}[x]$, $\text{Csc}[x]$, and $\text{Cot}[x]$; the inverse trigonometric functions $\text{ArcSin}[x]$, $\text{ArcCos}[x]$, $\text{ArcTan}[x]$, $\text{ArcSec}[x]$, $\text{ArcCsc}[x]$, and $\text{ArcCot}[x]$; the hyperbolic trigonometric functions $\text{Sinh}[x]$, $\text{Cosh}[x]$, and $\text{Tanh}[x]$; and their inverses $\text{ArcSinh}[x]$, $\text{ArcCosh}[x]$, and $\text{ArcTanh}[x]$. Generally, Mathematica tries to return an exact value unless otherwise specified with N .

Several examples of the natural logarithm and the exponential functions are given next. Mathematica often recognizes the properties associated with these functions and simplifies expressions accordingly.

Example 2.1.5 Entering

$N[\text{number}]$ or number/N returns approximations of number.

$\text{Exp}[x]$ computes e^x . Enter E to compute $e \approx 2.718$.

Exp[-5]/N

0.00673795

returns an approximation of $e^{-5} = 1/e^5$. Entering**Log[E³]**

3

computes $\ln e^3 = 3$. Entering

Log[x] computes $\ln x$. $\ln x$ and e^x are inverse functions ($\ln e^x = x$ and $e^{\ln x} = x$) and Mathematica uses these properties when simplifying expressions involving these functions.

Abs[x] returns the absolute value of x , $|x|$.

N[number] and **number//N** return approximations of number.

Exp[Log[π]]

π

computes $e^{\ln \pi} = \pi$. Entering

Abs[-5]

5

computes $|-5| = 5$. Entering

Abs[$\frac{3+2i}{2-9i}$]

$\sqrt{\frac{13}{85}}$

computes $|(3+2i)/(2-9i)|$. Entering

Cos[$\frac{\pi}{12}$]

$\frac{1+\sqrt{3}}{2\sqrt{2}}$

N[Cos[$\frac{\pi}{12}$]]

0.965926

computes the exact value of $\cos(\pi/12)$ and then an approximation. Although Mathematica cannot compute the exact value of $\tan 1000$, entering

N[Tan[1000]]

1.47032

returns an approximation of $\tan 1000$. Similarly, entering

N[ArcSin[1/3]]

0.339837

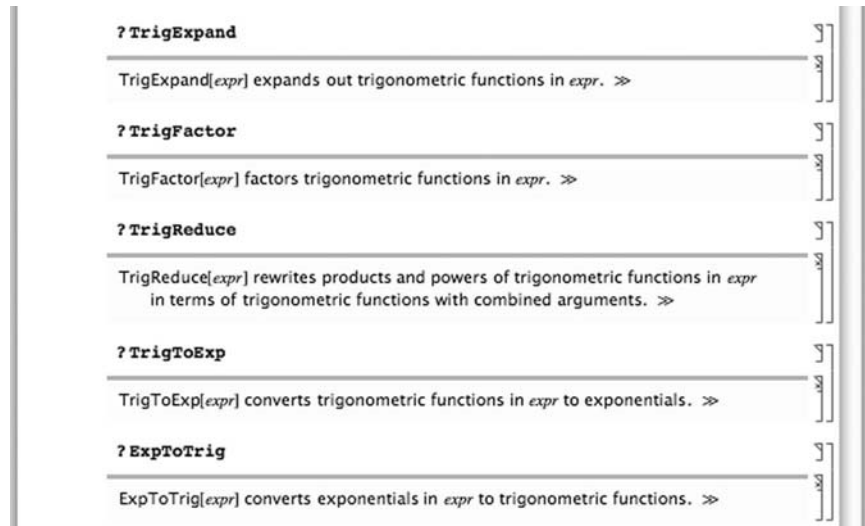
returns an approximation of $\sin^{-1}(1/3)$, and entering

ArcCos[2/3]//N

0.841069

returns an approximation of $\cos^{-1}(2/3)$.

Mathematica is able to apply many identities that relate the trigonometric and exponential functions using the functions **TrigExpand**, **TrigFactor**, **TrigReduce**, **TrigToExp**, and **ExpToTrig**.



Example 2.1.6 Mathematica does not automatically apply the identity $\sin^2 x + \cos^2 x = 1$.

Many of the algebraic manipulation commands can be accessed from the **Algebraic-Manipulation** palette.



$$\text{Cos}[x]^2 + \text{Sin}[x]^2$$

$$\cos^2 x + \sin^2 x$$

To apply the identity, we use **Simplify**. Generally, **Simplify[expression]** attempts to simplify expression.

$$\text{Simplify}[\text{Cos}[x]^2 + \text{Sin}[x]^2]$$

$$1$$

Use **TrigExpand** to multiply expressions or to rewrite trigonometric functions. In this case, entering

$$\text{TrigExpand}[\text{Cos}[3x]]$$

$$\cos^3 x - 3\cos x \sin^2 x$$

writes $\cos 3x$ in terms of trigonometric functions with argument x . We use the **TrigReduce** function to convert products to sums.

$$\text{TrigReduce}[\text{Sin}[3x]\text{Cos}[4x]]$$

$$\frac{1}{2}(-\sin x + \sin 7x)$$

We use **TrigExpand** to write

$$\text{TrigExpand}[\text{Cos}[2x]]$$

$$\cos^2 x - \sin^2 x$$

in terms of trigonometric functions with argument x . We use **ExpToTrig** to convert exponential expressions to trigonometric expressions.

ExpToTrig[1/2(Exp[x] + Exp[-x])]

Cosh[x]

Similarly, we use TrigToExp to convert trigonometric expressions to exponential expressions.

TrigToExp[Sin[x]]

$\frac{1}{2}ie^{-ix} - \frac{1}{2}ie^{ix}$

Usually, you can use Simplify to apply elementary identities.

Simplify[Tan[x]^2 + 1]

Sec[x]^2

A Word of Caution

Remember that there are certain ambiguities in traditional mathematical notation. For example, the expression $\sin^2(\pi/6)$ is usually interpreted to mean “compute $\sin(\pi/6)$ and square the result.” That is, $\sin^2(\pi/6) = [\sin(\pi/6)]^2$. The symbol sin is not being squared; the number $\sin(\pi/6)$ is squared. With Mathematica, we must be especially careful and follow the standard order of operations exactly, especially when using **InputForm**. We see that entering



```
Sin[Pi/6]^2
```

$$\frac{1}{4}$$

computes $\sin^2(\pi/6) = [\sin(\pi/6)]^2$, whereas



```
Sin^2[Pi/6]
```

$$\sin^2\left[\frac{\pi}{6}\right]$$

raises the symbol Sin to the power 2 $\left[\frac{\pi}{6}\right]$. Mathematica interprets



```
Sin[Pi/6]^2
```

$$\frac{1}{4}$$

to be the product of the symbols $\sin^2$ and $\pi/6$. However, using **TraditionalForm** we are able to evaluate $\sin^2(\pi/6) = [\sin(\pi/6)]^2$ with Mathematica using conventional mathematical notation.



```
Sin^2[Pi/6]
```

$$\frac{1}{4}$$

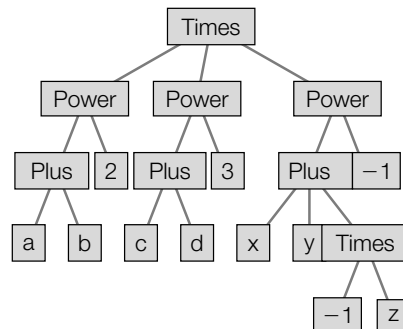
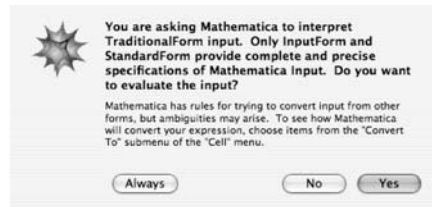


FIGURE 2.1

Visualizing the order in which Mathematica carries out a sequence of operations

Be aware, however, that traditional mathematical notation does contain certain ambiguities and Mathematica may not return the result you expect if you enter input using **TraditionalForm** unless you are especially careful to follow the standard order of operations, as the following warning message indicates.



Example 2.1.7 As stated, Mathematica follows the order of operations exactly. To see how Mathematica performs a calculation, **TreeForm** presents the sequence graphically. For example, for the calculation $\frac{(a+b)^2(c+d)^3}{x+y-z}$, **TreeForm** gives us the results shown in Figure 2.1.

```
Clear[a, b, c, d, x, y, z]
TreeForm[(a + b)^2(c + d)^3/(x + y - z)]
```

2.2 EXPRESSIONS AND FUNCTIONS: ELEMENTARY ALGEBRA

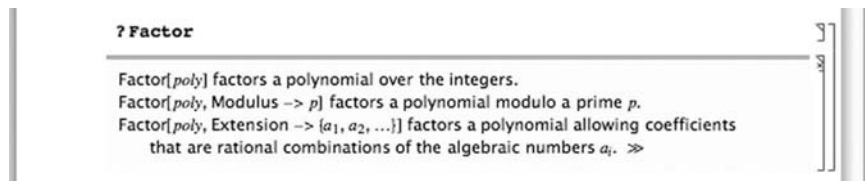
2.2.1 Basic Algebraic Operations on Expressions

Expressions involving unknowns are entered in the same way as numbers. Mathematica performs standard algebraic operations on mathematical expressions. For example, the commands

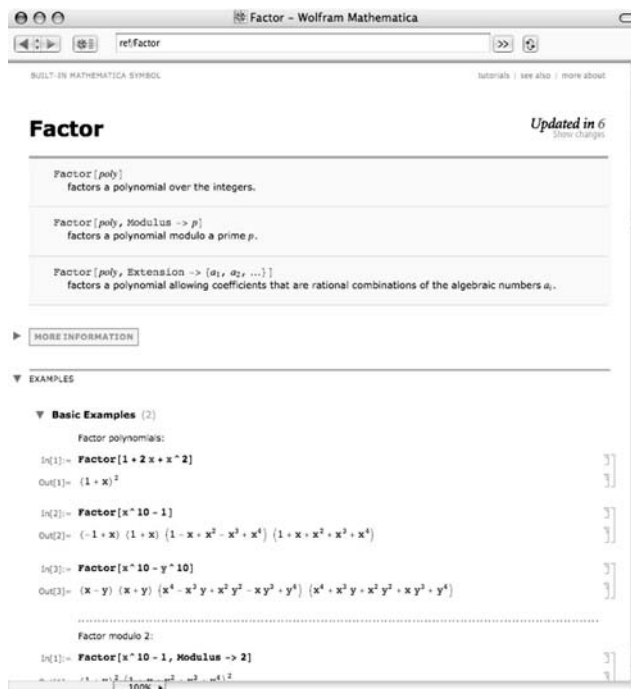
1. `Factor[expression]` factors expression;
2. `Expand[expression]` multiplies expression;

3. Together[expression] writes expression as a single fraction; and
4. Simplify[expression] performs basic algebraic manipulations on expression and returns the simplest form it finds.

For basic information about any of these commands (or any other), enter ?command as we do here for Factor,



or access the **Help Browser** as we do here for Factor.



When entering expressions, be sure to include a space or * between variables to denote multiplication.

Example 2.2.1 (a) Factor the polynomial $12x^2 + 27xy - 84y^2$. (b) Expand the expression $(x + y)^2(3x - y)^3$. (c) Write the sum $\frac{2}{x^2} - \frac{x^2}{2}$ as a single fraction.

Solution The result obtained with **Factor** indicates that $12x^2 + 27xy - 84y^2 = 3(4x - 7y)(x + 4y)$. When typing the command, be sure to include a space, or *, between the x and y terms to denote multiplication. **xy** represents an expression, whereas **x y** or **x*y** denotes x multiplied by y.

Factor[**12x² + 27xy - 84y²**]

$3(4x - 7y)(x + 4y)$

We use **Expand** to compute the product $(x + y)^2(3x - y)^3$ and **Together** to express

$\frac{2}{x^2} - \frac{x^2}{2}$ as a single fraction.

Expand[(**x + y**)²(**3x - y**)³]

$27x^5 + 27x^4y - 18x^3y^2 - 10x^2y^3 + 7xy^4 - y^5$

Together [$\frac{2}{x^2} - \frac{x^2}{2}$]

$\frac{4 - x^4}{2x^2}$

Factor[**x² - 3**] returns $x^2 - 3$.

To factor an expression such as $x^2 - 3 = x^2 - (\sqrt{3})^2 = (x - \sqrt{3})(x + \sqrt{3})$, use **Factor** with the **Extension** option.

Factor[**x² - 3, Extension → {Sqrt[3]}**]

$-(\sqrt{3} - x)(\sqrt{3} + x)$

Similarly, use **Factor** with the **Extension** option to factor expressions such as $x^2 + 1 = x^2 - i^2 = (x + i)(x - i)$.

Factor[**x² + 1**]

$1 + x^2$

Factor[**x² + 1, Extension → {I}**]

$(-i + x)(i + x)$

Mathematica does not automatically simplify $\sqrt{x^2}$ to the expression x

Sqrt[**x²**]

$\sqrt{x^2}$

because without restrictions on x , $\sqrt{x^2} = |x|$. The command **PowerExpand**[expression] simplifies expression assuming that all variables are positive. Alternatively, you can use **Assumptions** to tell Mathematica to assume that $x > 0$.

PowerExpand[**Sqrt**[**x²**]]

x

Simplify[**Sqrt**[**x²**], **Assumptions → x > 0**]

x

Thus, entering

Simplify[Sqrt[a^2b^4]]
 $\sqrt{a^2b^4}$

returns $\sqrt{a^2b^4}$, but entering

PowerExpand[Sqrt[a^2b^4]]
 ab^2

Simplify[Sqrt[a^2b^4], Assumptions → {a > 0, b > 0}]
 ab^2

returns ab^2 .

In general, a space is not needed between a number and a symbol to denote multiplication when a symbol follows a number. That is, 3dog means “3 times variable dog,” dog3 is a variable with name dog3. Mathematica interprets 3 dog, dog*3, and dog 3 as “3 times variable dog.” However, when multiplying two variables, either include a space or * between the variables.

1. cat dog means “variable cat times variable dog.”
2. cat*dog means “variable cat times variable dog.”
3. But, catdog is interpreted as a variable catdog.

The command Apart[expression] computes the partial fraction decomposition of expression; Cancel[expression] factors the numerator and denominator of expression and then reduces expression to lowest terms.

Example 2.2.2 (a) Determine the partial fraction decomposition of $\frac{1}{(x-3)(x-1)}$. (b) Simplify $\frac{x^2-1}{x^2-2x+1}$.

Solution Apart is used to see that $\frac{1}{(x-3)(x-1)} = \frac{1}{2(x-3)} - \frac{1}{2(x-1)}$. Then, Cancel is used to find that $\frac{x^2-1}{x^2-2x+1} = \frac{(x-1)(x+1)}{(x-1)^2} = \frac{x+1}{x-1}$. In this calculation, we have assumed that $x \neq 1$, an assumption made by Cancel but not by Simplify.

Apart $\left[\frac{1}{(x-3)(x-1)}\right]$
 $\frac{1}{2(-3+x)} - \frac{1}{2(-1+x)}$
Cancel $\left[\frac{x^2-1}{x^2-2x+1}\right]$
 $\frac{1+x}{-1+x}$

In addition, Mathematica has several built-in functions for manipulating parts of fractions.

1. `Numerator[fraction]` yields the numerator of fraction.
2. `ExpandNumerator[fraction]` expands the numerator of fraction.
3. `Denominator[fraction]` yields the denominator of fraction.
4. `ExpandDenominator[fraction]` expands the denominator of fraction.

Example 2.2.3 Given $\frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4}$, (a) factor both the numerator and denominator; (b) reduce $\frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4}$ to lowest terms; and (c) find the partial fraction decomposition of $\frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4}$.

Solution The numerator of $\frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4}$ is extracted with `Numerator`. We then use `Factor` together with `%`, which is used to refer to the most recent output, to factor the result of executing the `Numerator` command.

```
Numerator  $\left[ \frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4} \right]$ 
-2 - x + 2x2 + x3
```

```
Factor[%]
(-1 + x)(1 + x)(2 + x)
```

Similarly, we use `Denominator` to extract the denominator of the fraction. Again, `Factor` together with `%` is used to factor the previous result, which corresponds to the denominator of the fraction.

```
Denominator  $\left[ \frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4} \right]$ 
-4 - 4x + x2 + x3
```

```
Factor[%]
(-2 + x)(1 + x)(2 + x)
```

`Cancel` is used to reduce the fraction to lowest terms.

```
Cancel  $\left[ \frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4} \right]$ 
 $\frac{-1 + x}{-2 + x}$ 
```

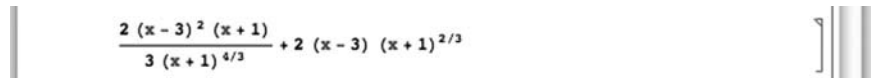
Finally, `Apart` is used to find its partial fraction decomposition.

```
Apart  $\left[ \frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4} \right]$ 
1 +  $\frac{1}{-2 + x}$ 
```

You can also take advantage of the **AlgebraicManipulation** palette, which is accessed by going to **Palettes** under the Mathematica menu, followed by **AlgebraicManipulation**, to evaluate expressions.

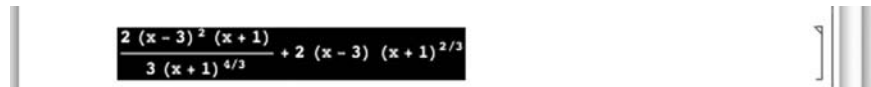
Example 2.2.4 Simplify $\frac{2(x-3)^2(x+1)}{3(x+1)^{4/3}} + 2(x-3)(x+1)^{2/3}$.

Solution First, we type the expression.



$$\frac{2 (x - 3)^2 (x + 1)}{3 (x + 1)^{4/3}} + 2 (x - 3) (x + 1)^{2/3}$$

Then, select the expression.



$$\frac{2 (x - 3)^2 (x + 1)}{3 (x + 1)^{4/3}} + 2 (x - 3) (x + 1)^{2/3}$$

Move the cursor to the palette and click on **Simplify**. Mathematica simplifies the expression.



$$\frac{8 (-3 + x) x}{3 (1 + x)^{1/3}}$$

2.2.2 Naming and Evaluating Expressions

In Mathematica, objects can be named. Naming objects is convenient: We can avoid typing the same mathematical expression repeatedly (as we did in Example 2.2.3) and named expressions can be referenced throughout a notebook or Mathematica session. Every Mathematica object can be named—expressions, functions, graphics, and so on can be named with Mathematica. Objects are named by using a single equals sign (=).

Because every built-in Mathematica function begins with a capital letter, we adopt the convention that *every* mathematical object we name in this text will begin with a *lowercase* letter. Consequently, we will be certain to avoid any possible ambiguity with any built-in Mathematica objects.

With Mathematica 6, the default option is to display *known* objects in black and *unknown* objects in blue. Thus, in the following screen shot,

```

Fraction x y
Apart
apart
2 Pi pi π
7 Plot Expand Cancel
E e E

```

Fraction, x, y, apart, pi, e, and E are in blue; Apart, 2, Pi, π , 7, Plot, Expand, Cancel, and E are in black.

To automatically update named variables, Dynamic[x] returns the current value of x .

Thus, Dynamic[x] returns dog.

```

x = dog
dog
Dynamic[x]
dog

```

However, when we enter $x = 7$ afterwards, Dynamic[x] is automatically updated to the new value of x .

```

x = dog
dog
Dynamic[x]
7
x = 7
7

```

Expressions are easily evaluated using ReplaceAll, which is abbreviated with /. and obtained by typing a backslash (/) followed by a period (.), together with Rule, which is abbreviated with -> and obtained by typing a forward slash (/) followed by a greater than sign (>). For example, entering the command

```
x^2 /. x -> 3
```

returns the value of the expression x^2 if $x = 3$. Note, however, that this does not assign the symbol x the value 3: entering $x=3$ assigns x the value 3.

Example 2.2.5 Evaluate $\frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4}$ if $x = 4$, $x = -3$, and $x = 2$.

Solution To avoid retyping $\frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4}$, we define `fraction` to be $\frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4}$. Of course, you can simply copy and paste this expression if you neither want to name it nor retype it.

If you include a semicolon (;) at the end of the command, the resulting output is suppressed.

$$\text{fraction} = \frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4}$$

$$\frac{-2 - x + 2x^2 + x^3}{-4 - 4x + x^2 + x^3}$$

`/.` is used to evaluate `fraction` if $x = 4$ and then if $x = -3$.

`fraction/.x->4`

$$\frac{3}{2}$$

`fraction/.x->-3`

$$\frac{4}{5}$$

When we try to replace each x in `fraction` by 2, we see that the result is undefined: division by 0 is always undefined.

```
fraction /. x -> 2
```

```
Power::infy : Infinite expression 1/0 encountered. >>
```

```
Infinity::indet : Indeterminate expression 0 ComplexInfinity encountered. >>
```

```
Indeterminate
```

However, when we use `Cancel` to first simplify and then use `ReplaceAll` to evaluate,

```
fraction2 = Cancel[fraction]
```

$$\frac{-1 + x}{-2 + x}$$

```
fraction2 /. x -> 2
```

$$\frac{3}{4}$$

we see that the result is $3/4$. The result indicates that $\lim_{x \rightarrow -2} \frac{x^3 + 2x^2 - x - 2}{x^3 + x^2 - 4x - 4} = \frac{3}{4}$. We confirm this result with `Limit`.

```
Limit[fraction, x -> -2]
```

$$\frac{3}{4}$$

Generally, `Limit[f[x], x->a]` attempts to compute $\lim_{x \rightarrow a} f(x)$. The `Limit` function is discussed in more detail in the next chapter.

2.2.3 Defining and Evaluating Functions

It is important to remember that functions, expressions, and graphics can be named anything that is not the name of a built-in Mathematica function or command. As previously indicated, every built-in Mathematica object begins with a capital letter so every user-defined function, expression, or other object in this text will be assigned a name using lowercase letters exclusively. This way, the possibility of conflicting with a built-in Mathematica command or function is completely eliminated. Because definitions of functions and names of objects are frequently modified, we introduce the command `Clear`. `Clear[expression]` clears all definitions of expression, if any. You can see if a particular symbol has a definition by entering `?symbol`.

In Mathematica, an elementary function of a single variable, $y = f(x) = \text{expression in } x$, is typically defined using the form

$$f[x_] = \text{expression in } x \text{ or } f[x_] := \text{expression in } x.$$

Notice that when you first define a function, you must always enclose the argument in square brackets (`[...]`) and place an underline (or blank) “`_`” after the argument on the left-hand side of the equals sign in the definition of the function.

Example 2.2.6 Entering

$$f[x_] = x/(x^2 + 1)$$

$$\frac{x}{1 + x^2}$$

defines and computes $f(x) = x/(x^2 + 1)$. Entering

$$f[3]$$

$$\frac{3}{10}$$

computes $f(3) = 3/(3^2 + 1) = 3/10$. Entering

$$f[a]$$

$$\frac{a}{1 + a^2}$$

computes $f(a) = a/(a^2 + 1)$. Entering

$$f[3 + h]$$

$$\frac{3 + h}{1 + (3 + h)^2}$$

computes $f(3 + h) = (3 + h)/((3 + h)^2 + 1)$. Entering

$$n1 = \text{Simplify}[(f[3 + h] - f[3])/h]$$

$$-\frac{8 + 3h}{10(10 + 6h + h^2)}$$

computes and simplifies $\frac{f(3+b) - f(3)}{b}$ and names the result n1. Entering

n1/.h→0

$$-\frac{2}{25}$$

evaluates n1 if $b = 0$. Entering

n2 = Together[(f[a + h] - f[a])/h]

$$\frac{1 - a^2 - ah}{(1 + a^2)(1 + a^2 + 2ah + h^2)}$$

computes and simplifies $\frac{f(a+b) - f(a)}{b}$ and names the result n2. Entering

n2/.h→0

$$\frac{1 - a^2}{(1 + a^2)^2}$$

evaluates n2 if $b = 0$.

Often, you will need to evaluate a function for the values in a **list**,

$$\text{list} = \{a_1, a_2, a_3, \dots, a_n\}.$$

Once $f(x)$ has been defined, Map [f,list] returns the list

$$\{f(a_1), f(a_2), f(a_3), \dots, f(a_n)\}.$$

Also,

1. Table [f[n], {n, n1, n2}] returns the list

$$\{f(n_1), f(n_1 + 1), f(n_1 + 2), \dots, f(n_2)\}.$$

2. Table [{n, f[n]},{n, n1, n2}] returns the list of ordered pairs

$$\{(n_1, f(n_1)), (n_1 + 1, f(n_1 + 1)), (n_1 + 2, f(n_1 + 2)), \dots, (n_2, f(n_2))\}.$$

The Table function will be discussed in more detail as needed.

Example 2.2.7 Entering

Clear[h]

h[t_] = (1 + t)^(1/t);

h[1]

2

defines $b(t) = (1 + t)^{1/t}$ and then computes $b(1) = 2$. Because division by 0 is always undefined, $b(0)$ is undefined.

h[0]

Power::infy: Infinite expression $\frac{1}{0}$ encountered. >>

Indet::indet: Indeterminate expression 1^{ComplexInfinity} encountered. >>

Indeterminate

`RandomReal[{a, b}]`
returns a random
real number bet-
ween a and b ;
`RandomReal[{a, b}, n]`
returns n random real
numbers between a
and b .

However, $b(t)$ is defined for all $t > 0$. In the following, we use `RandomReal` together with `Table` to generate six random numbers “close” to 0 and name the resulting list `t1`. Because we are using `RandomReal`, your results will almost certainly differ from those here.

```
t1 = Table[RandomReal[{0, 10^-n}], {n, 0, 5}]
{0.457711, 0.0446146, 0.00848021,
0.000465453, 0.0000566835, 1.6690247776250502*^-6}
```

We then use `Map` to compute $b(t)$ for each of the values in the list `t1`.

```
Map[h, t1]
{2.27817, 2.66002, 2.70684, 2.71765, 2.7182, 2.71828}
```

From the result, we might correctly deduce that $\lim_{t \rightarrow 0^+} (1+t)^{1/t} = e$.

In each of these cases, do not forget to include the blank (or underline) (`_`) on the left-hand side of the equals sign in the definition of each function. Remember to always include arguments of functions in square brackets.

Example 2.2.8 Entering

Including a semicolon
at the end of a
command suppresses
the resulting output.

```
Clear[f]
f[0] = 1;
f[1] = 1;
f[n_] := f[n - 1] + f[n - 2]
```

defines the recursively defined function defined by $f(0) = 1$, $f(1) = 1$, and $f(n) = f(n-1) + f(n-2)$. For example, $f(2) = f(1) + f(0) = 1 + 1 = 2$; $f(3) = f(2) + f(1) = 2 + 1 = 3$. We use `Table` to create a list of ordered pairs $(n, f(n))$ for $n = 0, 1, \dots, 10$.

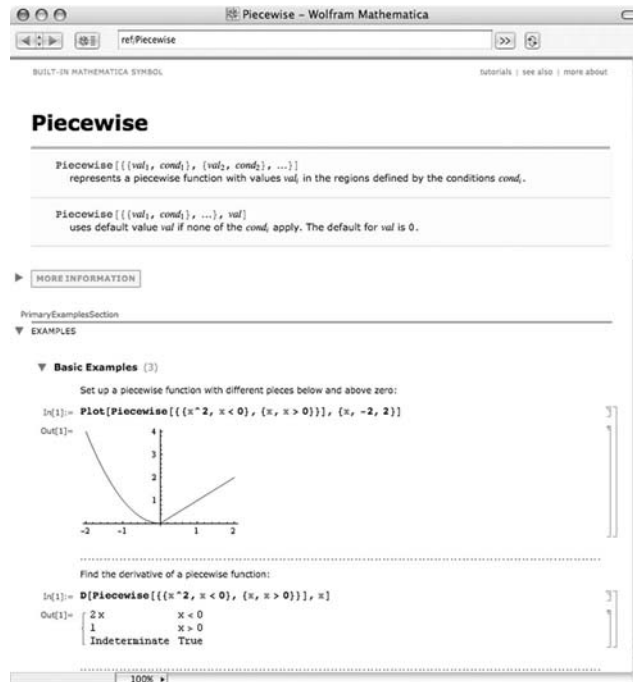
```
Table[{n, f[n]}, {n, 0, 10}]
{{0, 1}, {1, 1}, {2, 2}, {3, 3}, {4, 5},
{5, 8}, {6, 13}, {7, 21}, {8, 34}, {9, 55}, {10, 89}}
```

The f_n we have
defined here
returns the
Fibonacci number
 F_n . `Fibonacci[n]`
also returns the n th
Fibonacci number.

In the preceding examples, the functions were defined using each of the forms `f[x.] := ...` and `f[x_] = ...`. As a practical matter, when defining “routine” functions with domains consisting of sets of real numbers and ranges consisting of sets of real numbers, either form can be used. Defining a function using the form `f[x_] = ...` instructs Mathematica to define `f` and then compute and return `f[x]` (**immediate assignment**); defining a function using the form `f[x.] := ...` instructs Mathematica to define `f`. In this case, `f[x]` is not computed and, thus, Mathematica returns no output (**delayed assignment**). The form `f[x.] := ...` should be used when Mathematica cannot evaluate `f[x]` unless `x` is a particular value, as with recursively defined functions or piecewise-defined functions, which we will discuss soon.

Generally, if attempting to define a function using the form $f[x]=\dots$ produces one or more error messages, use the form $f[x]:= \dots$ instead.

To define piecewise-defined functions, we usually use Condition ($/;$) as illustrated in the following example. In simple situations, we take advantage of Piecewise.



Example 2.2.9 Entering

```
Clear[f]
f[t_]:=Sin[1/t]/; t > 0
```

defines $f(t) = \sin(1/t)$ for $t > 0$. Entering

```
f[1/(10Pi)]
0
```

is evaluated because $1/(10\pi) > 0$. However, both of the following commands are returned unevaluated. In the first case, -1 is not greater than 0 ($f(t)$ is not defined for $t \leq 0$). In the second case, Mathematica does not know the value of a so it cannot determine if it is or is not greater than 0 .

```
f[-1]
f[-1]
```

f[a]

f[a]

Entering

f[t_]:= -t; t ≤ 0

defines $f(t) = -t$ for $t \leq 0$. Now, the domain of $f(t)$ is all real numbers. That is, we have defined the piecewise-defined function

$$f(t) = \begin{cases} \sin(1/t), & t > 0 \\ -t, & t \leq 0 \end{cases}.$$

We can now evaluate $f(t)$ for any real number t .

f[2/(5Pi)]

1

f[0]

0

f[-10]

10

However, $f(a)$ still returns unevaluated because Mathematica does not know if $a \leq 0$ or if $a > 0$.

f[a]

f[a]

Recursively defined functions are handled in the same way. The following example shows how to define a periodic function.

Example 2.2.10 Entering**Clear[g]****g[x_]:= x;/ 0 ≤ x < 1****g[x_]:= 1;/ 1 ≤ x < 2****g[x_]:= 3 - x;/ 2 ≤ x < 3****g[x_]:= g[x - 3]/; x ≥ 3**

defines the recursively defined function $g(x)$. For $0 \leq x < 3$, $g(x)$ is defined by

$$g(x) = \begin{cases} x, & 0 \leq x < 1 \\ 1, & 1 \leq x < 2 \\ 3 - x, & 2 \leq x < 3. \end{cases}$$

For $x \geq 3$, $g(x) = g(x - 3)$. Entering

g[7]

1

computes $g(7) = g(4) = g(1) = 1$. We use **Table** to create a list of ordered pairs $(x, g(x))$ for 25 equally spaced values of x between 0 and 6.

Table[{x, g[x]}, {x, 0, 6, 6/24}]

$\{ \{0, 0\}, \{ \frac{1}{4}, \frac{1}{4} \}, \{ \frac{1}{2}, \frac{1}{2} \}, \{ \frac{3}{4}, \frac{3}{4} \}, \{1, 1\}, \{ \frac{5}{4}, 1 \},$
 $\{ \frac{3}{2}, 1 \}, \{ \frac{7}{4}, 1 \}, \{2, 1\}, \{ \frac{9}{4}, \frac{3}{4} \}, \{ \frac{5}{2}, \frac{1}{2} \}, \{ \frac{11}{4}, \frac{1}{4} \}, \{3, 0\},$
 $\{ \frac{13}{4}, \frac{1}{4} \}, \{ \frac{7}{2}, \frac{1}{2} \}, \{ \frac{15}{4}, \frac{3}{4} \}, \{4, 1\}, \{ \frac{17}{4}, 1 \}, \{ \frac{9}{2}, 1 \},$
 $\{ \frac{19}{4}, 1 \}, \{5, 1\}, \{ \frac{21}{4}, \frac{3}{4} \}, \{ \frac{11}{2}, \frac{1}{2} \}, \{ \frac{23}{4}, \frac{1}{4} \}, \{6, 0\} \}$

We will discuss additional ways to define, manipulate, and evaluate functions as needed. However, Mathematica's extensive programming language allows a great deal of flexibility in defining functions, many of which are beyond the scope of this text. These powerful techniques are discussed in detail in texts such as Gaylord, Kamin, and Wellin's *Introduction to Programming with Mathematica* [9], Gray's *Mastering Mathematica: Programming Methods and Applications* [12], and Maeder's *The Mathematica Programmer II* and *Programming in Mathematica* [15, 16].

2.3 GRAPHING FUNCTIONS, EXPRESSIONS, AND EQUATIONS

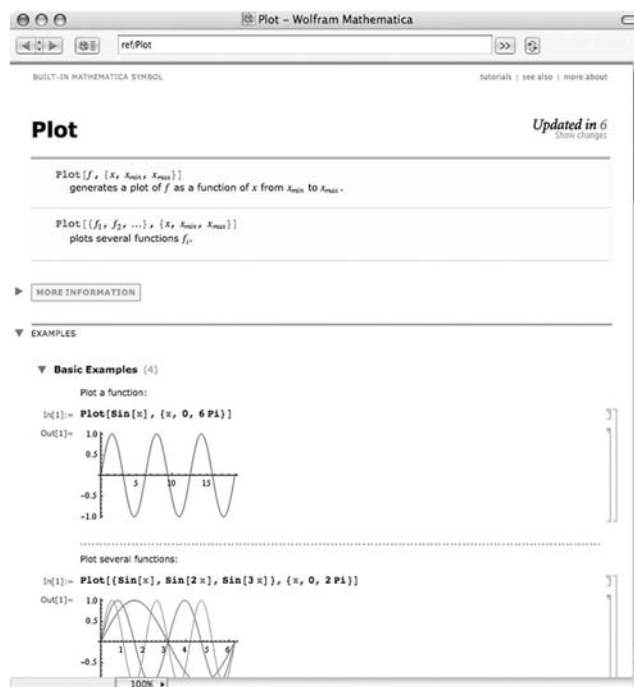
One of the best features of Mathematica is its graphics capabilities. In this section, we discuss methods of graphing functions, expressions, and equations and several of the options available to help graph functions.

2.3.1 Functions of a Single Variable

The commands

Plot[f[x], {x, a, b}] and **Plot[f[x], {x, a, x1, x2, ..., xn, b}]**

graph the function $y = f(x)$ on the intervals $[a, b]$ and $[a, x_1) \cup (x_1, x_2) \cup \dots \cup (x_n, b]$, respectively. Mathematica returns information about the basic syntax of the **Plot** command with **?Plot** or use the **Documentation Center** to obtain detailed information regarding **Plot**.



Remember that every Mathematica object can be assigned a name, including graphics. `Show[p1,p2,...,pn]` displays the graphics `p1`, `p2`, ..., `pn` together.

Example 2.3.1 Graph $y = \sin x$ for $-\pi \leq x \leq 2\pi$.

Solution Entering

p1 = Plot[Sin[x], {x, -Pi, 2Pi}]

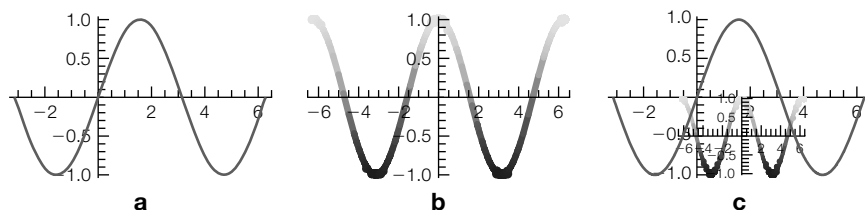
graphs $y = \sin x$ for $-\pi \leq x \leq 2\pi$ and names the result `p1`. The plot is shown in Figure 2.2(a). With

**p1b = Plot[Cos[x], {x, -2Pi, 2Pi},
ColorFunction → "ValentineTones",
PlotStyle → Thickness[.025]]**

we create a slightly thicker plot of $y = \cos x$ and shade the plot using the `ValentineTones` color gradient. See Figure 2.2(b).

`Show[p1,p2,...,pn]` shows the graphics `p1`, ..., `pn`. You can also use `Show` to rerender graphics. Using `Show` with the `Epilog` option together with `Inset`, we place a small version of the cosine plot in the sine plot. See Figure 2.2(c).

**p1c = Show[p1,
Epilog → Inset[p1b,{Pi/2, -1/2}, Automatic, 5]]**

**FIGURE 2.2**

(a) $y = \sin x$ for $-\pi \leq x \leq 2\pi$. (b) A “reddish” plot of $y = \cos x$ for $-2\pi \leq x \leq 2\pi$.
 (c) Combining two graphics with `Epilog` and `Inset`

Multiple graphics can be shown in rows, columns, or grids using `GraphicsRow`, `GraphicsColumn`, or `GraphicsGrid`, respectively. Thus,

```
Show[GraphicsRow[{p1, p1b, p1c}]]
```

generates Figure 2.2.

Be careful when graphing functions with discontinuities. Often, Mathematica will catch discontinuities. In other cases, it does not and you might need to use the `Exclusions` option to generate a more accurate plot.

Example 2.3.2 Graph $s(t)$ for $0 \leq t \leq 5$, where $s(t) = 1$ for $0 \leq t < 1$ and $s(t) = 1 + s(t - 1)$ for $t \geq 1$.

Solution After defining $s(t)$,

```
s[t_] := 1;/; 0 ≤ t < 1
s[t_] := 1 + s[t - 1]/; t ≥ 1
```

we use `Plot` to graph $s(t)$ for $0 \leq t \leq 5$ in Figure 2.3(a).

```
p1 = Plot[s[t], {t, 0, 5}, AspectRatio → Automatic]
```

Of course, Figure 2.3(a) is not completely precise: Vertical lines are never the graphs of functions. In this case, discontinuities occur at $t = 1, 2, 3, 4$, and 5 . If we were to redraw the figure by hand, we would erase the vertical line segments and then for emphasis place open dots at $(1, 1)$, $(2, 2)$, $(3, 3)$, $(4, 4)$, and $(5, 5)$ and then closed dots at $(1, 2)$, $(2, 3)$, $(3, 4)$, $(4, 5)$, and $(5, 6)$. In cases like this in which `Plot` does not automatically detect discontinuities, you can specify them with `Exclusions`. See Figure 2.3(b).

```
p2 = Plot[s[t], {t, 0, 5}, Exclusions → {1, 2, 3, 4}]
Show[GraphicsRow[{p1, p2}]]
```

To fine-tune graphics, use the **Drawing Tools** and **Graphics Inspector** palettes, which are accessed under **Graphics** in the menu. In this case, we add the closed dots at the left endpoints.

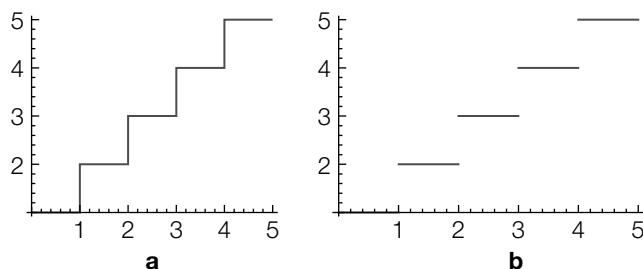
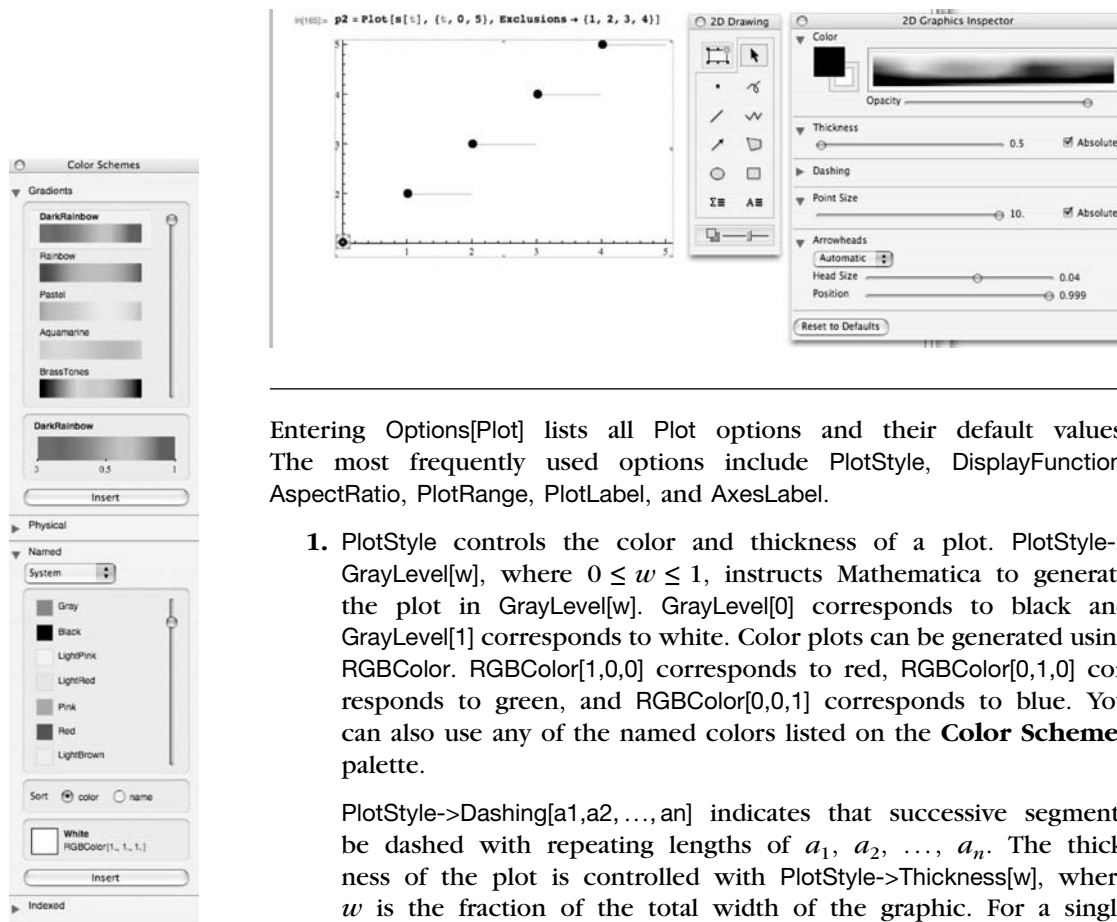


FIGURE 2.3

(a) $s(t) = 1 + s(t-1)$, $0 \leq t \leq 5$. (b) Catching the discontinuities



Entering `Options[Plot]` lists all `Plot` options and their default values. The most frequently used options include `PlotStyle`, `DisplayFunction`, `AspectRatio`, `PlotRange`, `PlotLabel`, and `AxesLabel`.

1. `PlotStyle` controls the color and thickness of a plot. `PlotStyle->GrayLevel[w]`, where $0 \leq w \leq 1$, instructs Mathematica to generate the plot in `GrayLevel[w]`. `GrayLevel[0]` corresponds to black and `GrayLevel[1]` corresponds to white. Color plots can be generated using `RGBColor`. `RGBColor[1,0,0]` corresponds to red, `RGBColor[0,1,0]` corresponds to green, and `RGBColor[0,0,1]` corresponds to blue. You can also use any of the named colors listed on the **Color Schemes** palette.

`PlotStyle->Dashing[a1,a2,...,an]` indicates that successive segments be dashed with repeating lengths of $a_1, a_2, \dots, a_n$. The thickness of the plot is controlled with `PlotStyle->Thickness[w]`, where w is the fraction of the total width of the graphic. For a single plot, the `PlotStyle` options are combined with `PlotStyle->{{option1, option2, ..., optionn}}`.

2. A plot is not displayed when the option `DisplayFunction->Identity` is included or when a semicolon (;) is included at the end of the command. Including the option `DisplayFunction->$DisplayFunction` in `Show` or `Plot` commands instructs Mathematica to display graphics.
3. The ratio of height to width of a plot is controlled by `AspectRatio`. The default is `1/GoldenRatio`. Generally, a plot is drawn to scale when the option `AspectRatio->Automatic` is included in the `Plot` or `Show` command.
4. `PlotRange` controls the horizontal and vertical axes. `PlotRange->{c,d}` specifies that the vertical axis displayed corresponds to the interval $c \leq y \leq d$, whereas `PlotRange->{{a,b},{c,d}}` specifies that the horizontal axis displayed corresponds to the interval $a \leq x \leq b$ and that the vertical axis displayed corresponds to the interval $c \leq y \leq d$.
5. `PlotLabel->"titleofplot"` labels the plot `titleofplot`.
6. `AxesLabel->{"xaxislabel","yaxislabel"}` labels the x -axis with `xaxislabel` and the y -axis with `yaxislabel`.

Example 2.3.3 Graph $y = \sin x$, $y = \cos x$, and $y = \tan x$ together with their inverse functions.

Solution In `p2` and `p3`, we use `Plot` to graph $y = \sin^{-1} x$ and $y = x$, respectively. Neither plot is displayed because we include a semicolon at the end of the `Plot` commands. `p1`, `p2`, and `p3` are displayed together with `Show` in Figure 2.4. The plot is shown to scale; the graph of $y = \sin x$ is in black, $y = \sin^{-1} x$ is in gray, and $y = x$ is dashed.

Be sure you have completed the previous example immediately before entering the following commands.

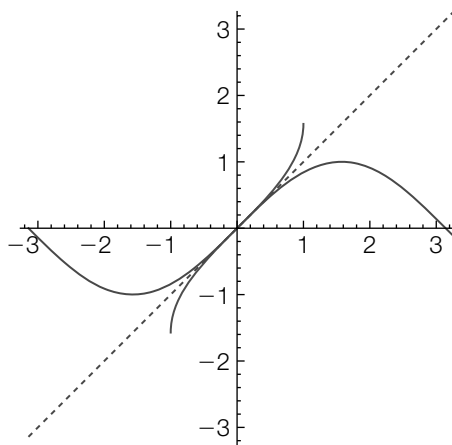


FIGURE 2.4

$y = \sin x$, $y = \sin^{-1} x$, and $y = x$

```

p2 = Plot[ArcSin[x], {x, -1, 1}, PlotStyle → GrayLevel[.3]];
p3 = Plot[x, {x, -Pi, 2Pi}, PlotStyle → Dashing[{.01}]];
p4 = Show[p1, p2, p3, PlotRange → {{-Pi, Pi}, {-Pi, Pi}}, AspectRatio →
Automatic]

```

The command `Plot[{f1[x], f2[x], ..., fn[x]}, {x, a, b}]` plots $f_1(x)$, $f_2(x)$, ..., $f_n(x)$ together for $a \leq x \leq b$. Simple `PlotStyle` options are incorporated with `PlotStyle->{option1, option2, ..., optionn}`, where `optioni` corresponds to the plot of $f_i(x)$. Multiple options are incorporated using `PlotStyle->{{options1}, {options2}, ..., {optionsn}}`, where `optionsi` are the options corresponding to the plot of $f_i(x)$.

In the following, we use `Plot` to graph $y = \cos x$, $y = \cos^{-1} x$, and $y = x$ together. The plot in Figure 2.5 is shown to scale; the graph of $y = \cos x$ is in black, $y = \cos^{-1} x$ is in gray, and $y = x$ is dashed.

```

r4 = Plot[{Cos[x], ArcCos[x], x}, {x, -Pi, Pi},
PlotStyle → {GrayLevel[0],
GrayLevel[.3], Dashing[{.01}]},
PlotRange → {-Pi, Pi}, AspectRatio → Automatic]

```

We use the same idea to graph $y = \tan x$, $y = \tan^{-1} x$, and $y = x$ in Figure 2.6

```

q4 = Plot[{Tan[x], ArcTan[x], x}, {x, -Pi, Pi},
PlotStyle → {GrayLevel[0],
GrayLevel[.3], Dashing[{.01}]},
PlotRange → {-Pi, Pi}, AspectRatio → Automatic]

```

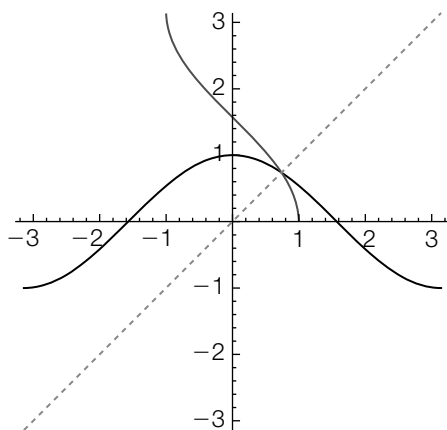


FIGURE 2.5

$y = \cos x$, $y = \cos^{-1} x$, and $y = x$

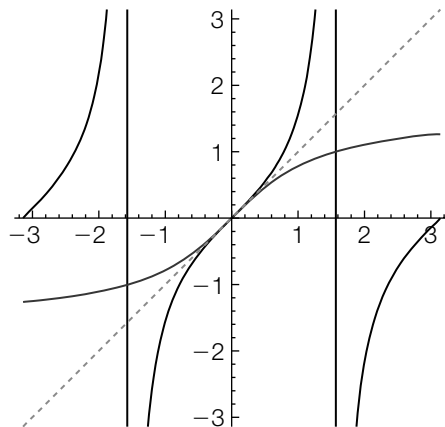


FIGURE 2.6

$y = \tan x$, $y = \tan^{-1} x$, and $y = x$

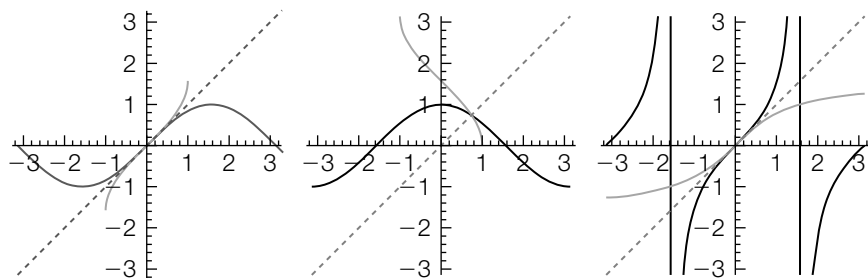


FIGURE 2.7

The elementary trigonometric functions and their inverses

Use `Show` together with `GraphicsRow` to display graphics in rectangular arrays. Entering

```
Show[GraphicsRow[{p4, r4, q4}]]
```

shows the three plots `p4`, `r4`, and `q4` in a row as shown in Figure 2.7.

The previous example illustrates the graphical relationship between a function and its inverse.

Example 2.3.4 (Inverse Functions). $f(x)$ and $g(x)$ are **inverse functions** if

$$f(g(x)) = g(f(x)) = x.$$

If $f(x)$ and $g(x)$ are inverse functions, their graphs are symmetric about the line $y = x$. The command

```
Composition[f1,f2,f3,...,fn,x]
```

computes the composition

$$(f_1 \circ f_2 \circ \cdots \circ f_n)(x) = f_1(f_2(\cdots(f_n(x)))).$$

For two functions $f(x)$ and $g(x)$, it is usually easiest to compute the composition $f(g(x))$ with `f[g[x]]` or `f[x]/g`.

Show that

$$f(x) = \frac{-1 - 2x}{-4 + x} \quad \text{and} \quad g(x) = \frac{4x - 1}{x + 2}$$

are inverse functions.

Solution After defining $f(x)$ and $g(x)$,

$f(x)$ and $g(x)$ are not returned because a semicolon is included at the end of each command.

$$f[x_] = \frac{-1 - 2x}{-4 + x};$$

$$g[x_] = \frac{4x - 1}{x + 2};$$

we compute and simplify the compositions $f(g(x))$ and $g(f(x))$. Because both results are x , $f(x)$ and $g(x)$ are inverse functions.

f[g[x]]

$$\frac{-1 - \frac{2(-1 + 4x)}{2 + x}}{-4 + \frac{-1 + 4x}{2 + x}}$$

Simplify[f[g[x]]]

x

Simplify[g[f[x]]]

x

To see that the graphs of $f(x)$ and $g(x)$ are symmetric about the line $y = x$, we use **Plot** to graph $f(x)$, $g(x)$, and $y = x$ together in Figure 2.8. Because **Tooltip** is being applied to the set of functions being plotted, you can identify each curve by sliding the cursor over the curve: when the cursor is placed over a curve, Mathematica displays its definition.

```
Plot[Tooltip[{f[x], g[x], f[g[x]]}], {x, -10, 10},
PlotStyle -> {GrayLevel[0], GrayLevel[.3],
Dashing[{.01]}], PlotRange -> {-10, 10},
AspectRatio -> Automatic]
```

In the plot, observe that the graphs of $f(x)$ and $g(x)$ are symmetric about the line $y = x$. The plot also illustrates that the domain and range of a function and its inverse are interchanged: $f(x)$ has domain $(-\infty, 4) \cup (4, \infty)$ and range $(-\infty, -2) \cup (-2, \infty)$; $g(x)$ has domain $(-\infty, -2) \cup (-2, \infty)$ and range $(-\infty, 4) \cup (4, \infty)$.

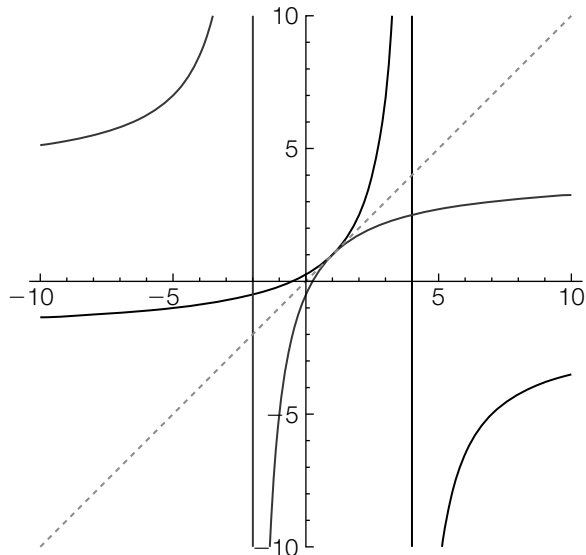


FIGURE 2.8

$f(x)$ in black, $g(x)$ in gray, and $y = x$ dashed

For repeated compositions of a function with itself, `Nest[f,x,n]` computes the composition

$$\underbrace{(f \circ f \circ f \circ \cdots \circ f)}_{n \text{ times}}(x) = \underbrace{(f(f(f \cdots)))}_{n \text{ times}}(x) = f^n(x).$$

Example 2.3.5 Graph $f(x)$, $f^{10}(x)$, $f^{20}(x)$, $f^{30}(x)$, $f^{40}(x)$, and $f^{50}(x)$ if $f(x) = \sin x$ for $0 \leq x \leq 2\pi$.

Solution After defining $f(x) = \sin x$,

```
f[x_] = Sin[x]
```

```
Sin[x]
```

we graph $f(x)$ in `p1` with `Plot`

```
p1 = Plot[f[x], {x, 0, 2Pi}];
```

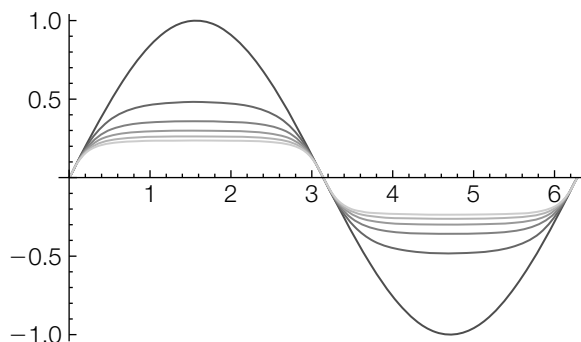
and then illustrate the use of `Nest` by computing $f^5(x)$.

```
Nest[f, x, 5]
```

```
Sin[Sin[Sin[Sin[Sin[x]]]]]
```

Next, we use `Table` together with `Nest` to create the list of functions

$$\{f^{10}(x), f^{20}(x), f^{30}(x), f^{40}(x), f^{50}(x)\}.$$

**FIGURE 2.9**

$f(x)$ in black; the graphs of $f^{10}(x)$, $f^{20}(x)$, $f^{30}(x)$, $f^{40}(x)$, and $f^{50}(x)$ are successively lighter—the graph of $f^{50}(x)$ is the lightest

`Table[f[i], {i, a, b, istep}]`
computes $f(i)$ for i
values from a to b
using increments of
 $istep$.

Because the resulting output is rather long, we include a semicolon at the end of the `Table` command to suppress the resulting output.

```
toplot = Table[Nest[f, x, n], {n, 10, 50, 10}];
```

We then graph the functions in `toplot` on the interval $[0, 2\pi]$ with `Plot`, applying the `Tooltip` function to the list being plotted so they can easily be identified. Last, we use `Show` to display `p1` and `p2` together in Figure 2.9.

```
p2 = Plot[Tooltip[toplot], {x, 0, 2Pi}];  
Show[p1, p2]
```

In the plot, we see that repeatedly composing sine with itself has a flattening effect on $y = \sin x$.

The command

```
ListPlot[{x1, y1}, {x2, y2}, ..., {xn, yn}]
```

plots the list of points $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. The size of the points in the resulting plot is controlled with the option `PlotStyle->PointSize[w]`, where w is the fraction of the total width of the graphic. For two-dimensional graphics, the default value is 0.008.

Remark 2.2 The command

```
ListPlot[y1, y2, ..., yn]
```

plots the list of points $\{(1, y_1), (2, y_2), \dots, (n, y_n)\}$.

Example 2.3.6 Graph $y = \frac{\sqrt{9-x^2}}{x^2-4}$.

Solution We use `Plot` to generate the basic graph of y shown in Figure 2.10(a).

p1 = Plot[Sqrt[9-x^2]/(x^2-4), {x, -5, 5}]

Observe that the domain of y is $[-3, -2) \cup (-2, 2) \cup (2, 3]$. A better graph of y is obtained by plotting y for $-3 \leq x \leq 3$ and shown in Figure 2.10(b). We then use the `PlotRange` option to specify that the displayed horizontal axis corresponds to $-7 \leq x \leq 7$ and that the displayed vertical axis corresponds to $-7 \leq y \leq 7$. The graph is drawn to scale because we include the option `AspectRatio->Automatic`. In this case, Mathematica does not generate any error messages. Mathematica uses a point-plotting scheme to generate graphs. Coincidentally, Mathematica does not sample $x = \pm 2$ and thus does not generate any error messages.

p2 = Plot[Sqrt[9-x^2]/(x^2-4), {x, -5, 5}, PlotRange -> {{-7, 7}, {-7, 7}}, AspectRatio -> Automatic]

To see the endpoints in the plot, we use `ListPlot` to plot the points $(-3, 0)$ and $(3, 0)$. The points are slightly enlarged in Figure 2.10(c) because we increase their size using `PointSize`.

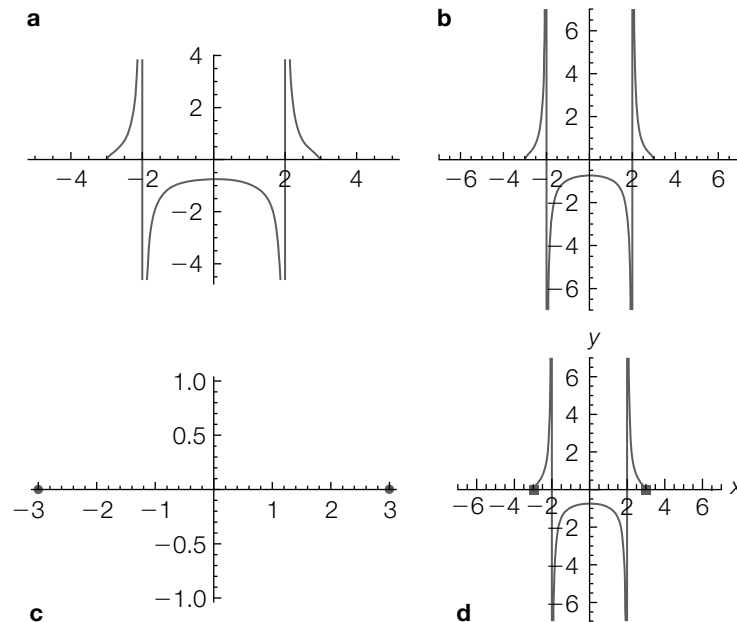


FIGURE 2.10

The four plots p1, p2, p3, and p4 combined into a single graphic

```
p3 = ListPlot[{{-3, 0}, {3, 0}}, PlotStyle -> PointSize[.02]]
```

Finally, we use `Show` to display `p2` and `p3` together in Figure 2.10(d), where we have labeled the axes using the `AxesLabel` option.

```
p4 = Show[p2, p3, AxesLabel -> {"x", "y"}]
```

The sequence of plots shown in Figure 2.10, which combines `p1`, `p2`, `p3`, and `p4` into a single graphic, is generated using `Show` together with `GraphicsGrid`.

```
Show[GraphicsGrid[{{p1, p2}, {p3, p4}}]]
```

When graphing functions involving odd roots, Mathematica's results may be surprising to the beginner. The key is to load the **RealOnly** package located at the Wolfram website first *or* remember that Mathematica follows the order of operations *exactly* and understand that without restrictions on x , $\sqrt{x^2} = |x|$.

Example 2.3.7 Graph $y = x^{1/3}(x-2)^{2/3}(x+1)^{4/3}$.

Solution Entering

```
p1 = Plot[x^(1/3)(x-2)^(2/3)(x+1)^(4/3),
{x, -3, 5}, PlotRange -> {-4, 4},
AspectRatio -> Automatic]
```

not does not produce the graph we expect (see Figure 2.11(a)) because many of us consider $y = x^{1/3}(x-2)^{2/3}(x+1)^{4/3}$ to be a real-valued function with domain $(-\infty, \infty)$. Generally, Mathematica does return a real number when computing the odd root of a negative number. For example, $x^3 = -1$ has three solutions:

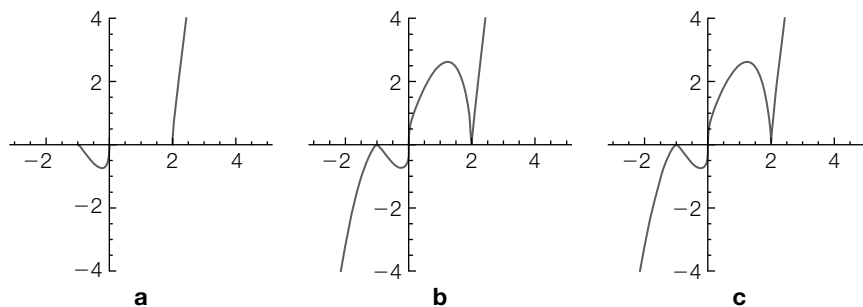


FIGURE 2.11

Three plots of $y = x^{1/3}(x-2)^{2/3}(x+1)^{4/3}$

Solve is discussed in more detail in the next section.

N[number] returns an approximation of number.

```
s1 = Solve[x^3 + 1 == 0]
{{x → -1}, {x → (-1)^(1/3)}, {x → -(-1)^(2/3)}}
```

```
N[s1]
{{x → -1.}, {x → 0.5 + 0.866025i}, {x → 0.5 - 0.866025i}}
```

When computing an odd root of a negative number, Mathematica has many choices (as illustrated above) and chooses a root with positive imaginary part—the result is not a real number.

```
(-1)^(1/3)/N
0.5 + 0.866025i
```

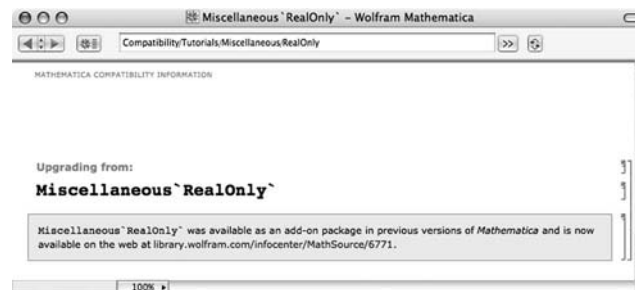
To obtain real values when computing odd roots of negative numbers, first

let $sign(x) = \begin{cases} x/|x|, & \text{if } x \neq 0, \\ 0, & \text{if } x = 0 \end{cases}$. Sign[x] returns *sign(x)*. Then, for the reduced

fraction n/m with m odd, $x^{n/m} = \begin{cases} sign(x)|x|^{n/m}, & \text{if } n \text{ is odd} \\ |x|^{n/m}, & \text{if } n \text{ is even} \end{cases}$. See Figure 2.11(b).

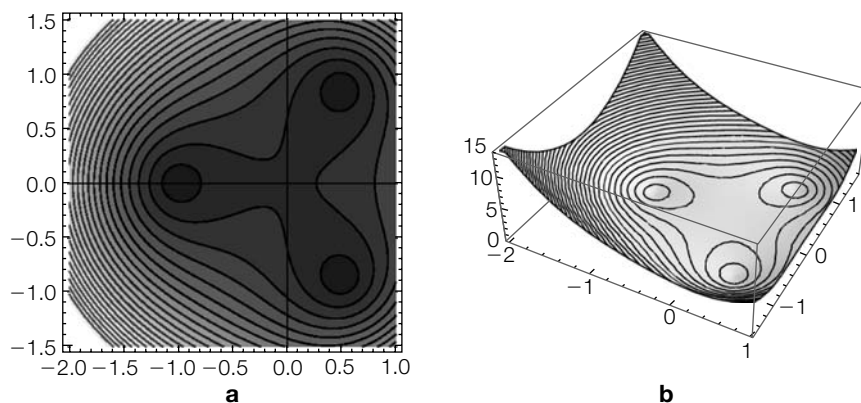
```
p2 = Plot[Sign[x]Abs[x]^(1/3)Abs[x - 2]^(2/3)Abs[x + 1]^(4/3),
{x, -3, 5}, PlotRange → {-4, 4}, AspectRatio → Automatic]
```

Alternatively, load the **RealOnly** package that is located in the **Miscellaneous** folder or directory if you have an older version of Mathematica or that can be downloaded from the Wolfram website if you only have version 6 or later.



After the **RealOnly** package has been loaded, reentering the **Plot** command produces the expected graph. See Figure 2.11c.

```
<< Miscellaneous`RealOnly`
p3 = Plot[x^(1/3)(x - 2)^(2/3)(x + 1)^(4/3), {x, -3, 5},
PlotRange → {-4, 4}, AspectRatio → Automatic]
Show[GraphicsRow[{p1, p2, p3}]]
```

**FIGURE 2.12**(a) Contour plot of $f(x,y)$, (b) 3D plot of $f(x,y)$

A comprehensive discussion of Mathematica's extensive graphics capabilities cannot be reasonably covered in a single text, so our approach is to address issues that might be of interest or present a different point of view to the novice. In the previous example, we saw that $x^3 + 1 = 0$ has three solutions, two of which are complex. To visualize this graphically, observe that the zeros of $z^3 + 1 = 0$ are the level curves of $f(x,y) = |(x+iy)^3 + 1|$ (x, y real) corresponding to 0. In a plot of $f(x,y)$, the solutions are the zeros. Soon, we will discuss ContourPlot and Plot3D. For now, we remark that

```
cp1 = ContourPlot[Abs[(x + Iy)^3 + 1], {x, -2, 1}, {y, -3/2, 3/2},
  Contours -> 30, Axes -> True]
p13d = Plot3D[Abs[(x + Iy)^3 + 1], {x, -2, 1}, {y, -3/2, 3/2},
  Axes -> True, PlotRange -> {0, 15}, MeshFunctions -> {#3 &}, Mesh -> 35]
Show[GraphicsRow[{cp1, p13d}]]
```

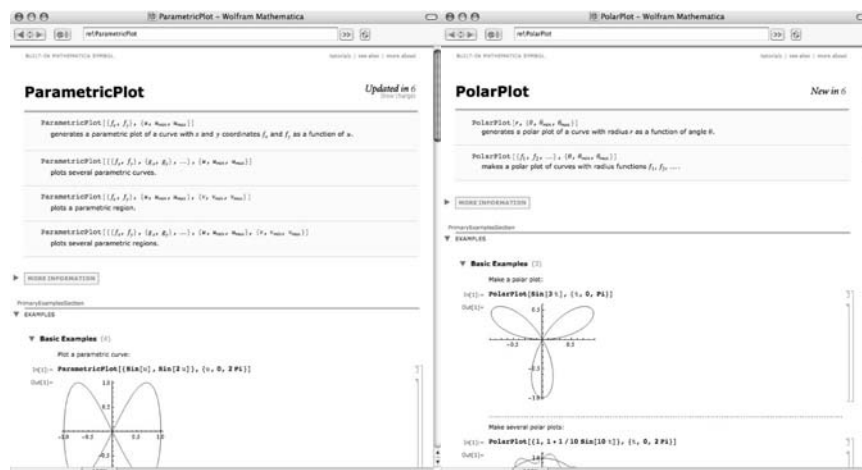
generates several level curves of $f(x,y)$ (Figure 2.12(a)) and a three-dimensional (3D) plot of $f(x,y)$ (Figure 2.12(b)) that help us see the zeros of the original equation. In the 3D plot, note how we use the MeshFunctions option to generate contours.

2.3.2 Parametric and Polar Plots in Two Dimensions

To graph the parametric equations $x = x(t)$, $y = y(t)$, $a \leq t \leq b$, use

ParametricPlot[[$x[t]$, $y[t]$], { t , a , b }]

ParametricPlot has the same options as Plot.



and to graph the polar function $r = r(\theta)$, $\alpha \leq \theta \leq \beta$, use

PolarPlot[r[theta], {theta, alpha, beta}].

Example 2.3.8 (The Unit Circle). The **unit circle** is the set of points (x, y) exactly 1 unit from the origin, $(0, 0)$, and, in rectangular coordinates, has equation $x^2 + y^2 = 1$. The unit circle is the classic example of a relation that is neither a function of x nor a function of y . The top half of the unit circle is given by $y = \sqrt{1 - x^2}$ and the bottom half is given by $y = -\sqrt{1 - x^2}$.

```
p1 = Plot[{Sqrt[1 - x^2], -Sqrt[1 - x^2]}, {x, -1, 1},
PlotRange -> {{-3/2, 3/2}, {-3/2, 3/2}},
AspectRatio -> Automatic];
```

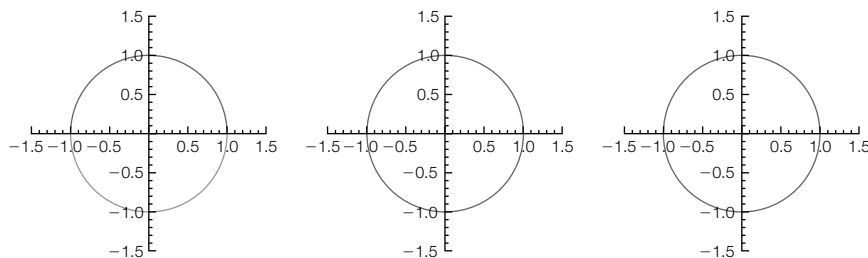
Each point (x, y) on the unit circle is a function of the angle, t , that subtends the x -axis, which leads to a parametric representation of the unit circle,

$$\begin{cases} x = \cos t, \\ y = \sin t, \end{cases} \quad 0 \leq t \leq 2\pi, \text{ which we graph with ParametricPlot.}$$

```
p2 = ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2Pi},
PlotRange -> {{-3/2, 3/2}, {-3/2, 3/2}},
AspectRatio -> Automatic];
```

Using the change of variables $x = r \cos t$ and $y = r \sin t$ to convert from rectangular to polar coordinates, a polar equation for the unit circle is $r = 1$. We use **PolarPlot** to graph $r = 1$.

```
p3 = PolarPlot[1, {t, 0, 2Pi},
PlotRange -> {{-3/2, 3/2}, {-3/2, 3/2}},
AspectRatio -> Automatic];
```

**FIGURE 2.13**

The unit circle generated with `Plot`, `ParametricPlot`, and `PolarPlot`

We display `p1`, `p2`, and `p3` side-by-side using `Show` together with `GraphicsRow` in Figure 2.13. Of course, they all look the same.

Show[GraphicsRow[{p1, p2, p3}]]

Example 2.3.9 Graph the parametric equations

$$\begin{cases} x = t + \sin 2t, \\ y = t + \sin 3t, \end{cases} \quad -2\pi \leq t \leq 2\pi.$$

Solution After defining x and y , we use `ParametricPlot` to graph the parametric equations in Figure 2.14(a).

```
x[t_] = t + Sin[2t];  
y[t_] = t + Sin[3t];  
p1 = ParametricPlot[{x[t], y[t]}, {t, -2Pi, 2Pi},  
    AspectRatio -> Automatic]
```

In Figure 2.14(b), we illustrate how to use the `PlotStyle` option to increase the thickness of the plot. Color is introduced using `ColorFunction` together with `ColorData`. We choose to use the `SolarColors` gradient to produce our plot.

```
x[t_] = t + Sin[2t];  
y[t_] = t + Sin[3t];  
p2 = ParametricPlot[{x[t], y[t]}, {t, -2Pi, 2Pi}, PlotStyle -> Thickness[.02],  
    AspectRatio -> Automatic, ColorFunction -> (ColorData  
        ["SolarColors"][#3]&)]
```

`ParametricPlot` can also be used to parametrically plot a region. In Figure 2.14(c), we plot $(r^2 x(t), r^2 y(t))$ for $-2\pi \leq t \leq 2\pi$ and $0 \leq r \leq 2$.

```
x[t_] = t + Sin[2t];  
y[t_] = t + Sin[3t];  
p3 = ParametricPlot[r^2{x[t], y[t]}, {t, -2Pi, 2Pi}, {r, 0, 2},
```

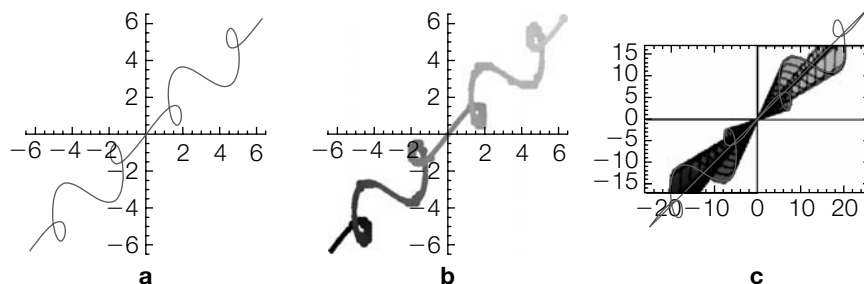


FIGURE 2.14

(a) $(x(t), y(t))$, $-2\pi \leq t \leq 2\pi$, (b) Adding some color and increasing the thickness, (c) Adding a second parameter

```
AspectRatio → Automatic, ColorFunction → (ColorData
["SolarColors"][#3]&)]
Show[GraphicsRow[{p1, p2, p3}]]
```

In the following example, the equations involve integrals.

Remark 2.3 Topics from calculus are discussed in Chapter 3. For now, we state that `Integrate[f[x], {x, a, b}]` attempts to evaluate $\int_a^b f(x) dx$.

Example 2.3.10 (Cornu Spiral). The **Cornu spiral** (or **clothoid**) (see [11] and [20]) has parametric equations

$$x = \int_0^t \sin\left(\frac{1}{2}u^2\right) du \quad \text{and} \quad y = \int_0^t \cos\left(\frac{1}{2}u^2\right) du.$$

Graph the Cornu spiral.

Solution We begin by defining x and y . Notice that Mathematica can evaluate these integrals, even though the results are in terms of the `FresnelS` and `FresnelC` functions, which are defined in terms of integrals:

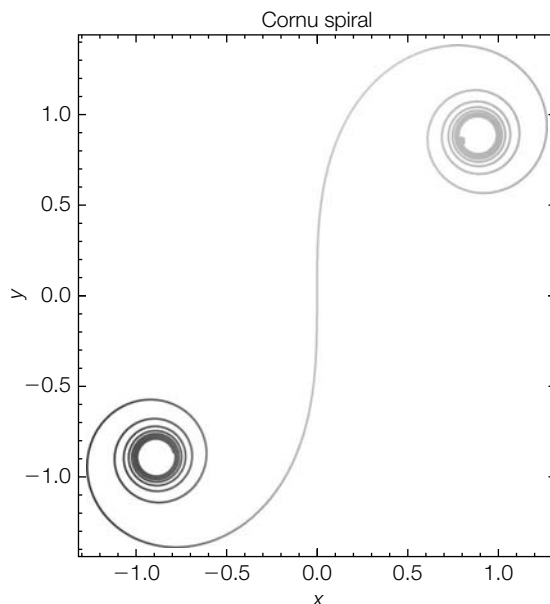
$$\text{FresnelS}[t] = \int_0^t \sin\left(\frac{\pi}{2}u^2\right) du \quad \text{and} \quad \text{FresnelC}[t] = \int_0^t \cos\left(\frac{\pi}{2}u^2\right) du.$$

```
x[t_] = Integrate[Sin[u^2/2], {u, 0, t}]
```

```
Sqrt[Pi]FresnelS[ $\frac{t}{\sqrt{\pi}}$ ]
```

```
y[t_] = Integrate[Cos[u^2/2], {u, 0, t}]
```

```
Sqrt[Pi]FresnelC[ $\frac{t}{\sqrt{\pi}}$ ]
```

**FIGURE 2.15**

The Cornu spiral

We use `ParametricPlot` to graph the Cornu spiral in Figure 2.15. The option `AspectRatio->Automatic` instructs Mathematica to generate the plot to scale; `PlotLabel->"Cornu spiral"` labels the plot.

```
ParametricPlot[{x[t], y[t]}, {t, -10, 10}, AspectRatio -> Automatic,  
PlotStyle -> Thickness[.01], PlotLabel -> "Cornuspiral",  
Frame -> True, FrameLabel -> {x, y},  
ColorFunction -> (ColorData["SouthwestColors"][#1]&)]
```

Observe that the graph of the polar equation $r = f(\theta)$, $\alpha \leq \theta \leq \beta$ is the same as the graph of the parametric equations

$$x = f(\theta) \cos \theta \quad \text{and} \quad y = f(\theta) \sin \theta, \quad \alpha \leq \theta \leq \beta$$

so both `ParametricPlot` and `PolarPlot` can be used to graph polar equations.

Example 2.3.11 Graph (a) $r = \sin(8\theta/7)$, $0 \leq \theta \leq 14\pi$; (b) $r = \theta \cos \theta$, $-19\pi/2 \leq \theta \leq 19\pi/2$; (c) ("The Butterfly") $r = e^{\cos \theta} - 2 \cos 4\theta + \sin^5(\theta/12)$, $0 \leq \theta \leq 24\pi$; and (d) ("The Lituus") $r^2 = 1/\theta$, $0.1 \leq \theta \leq 10\pi$.

Solution For (a) and (b) we use `ParametricPlot`. First define r and then use `ParametricPlot` to generate the graph of the polar curve. No graphics are displayed because we place a semicolon at the end of each command.

```
Clear[r]
r[θ_] = Sin[8θ/7];
pp1 = ParametricPlot[{r[θ]Cos[θ], r[θ]Sin[θ]},
  {θ, 0, 14Pi}, AspectRatio → Automatic];
```

For (b), we use the option `PlotRange->{{-30,30},{-30,30}}` to indicate that the range displayed on both vertical and horizontal axes corresponds to the interval $[-30, 30]$. To help (a) ensure that the resulting graphic appears “smooth,” we increase the number of points that Mathematica samples when generating the graph by including the option `PlotPoints->200`.

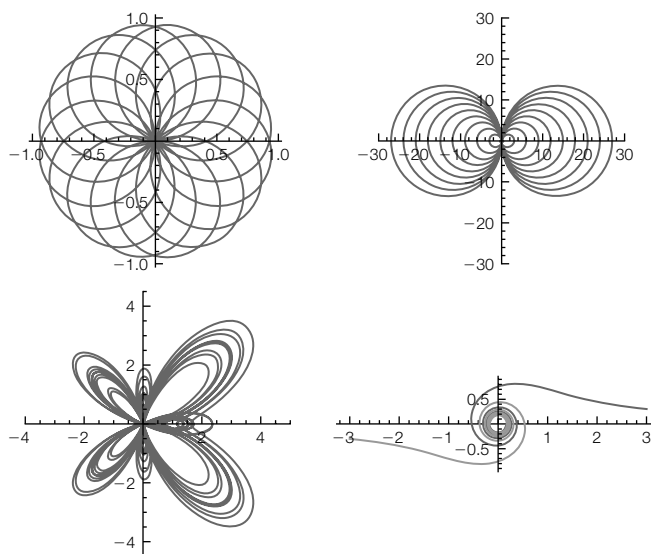
```
Clear[r]
r[θ_] = θCos[θ];
pp2 = ParametricPlot[{r[θ]Cos[θ], r[θ]Sin[θ]},
  {θ, -19π/2, 19π/2}, PlotRange → {{-30, 30}, {-30, 30}},
  AspectRatio → Automatic, PlotPoints → 200];
```

For (c) and (d), we use `PolarPlot`. Using standard mathematical notation, we know that $\sin^5(\theta/12) = (\sin(\theta/12))^5$. However, when defining r with Mathematica, be sure you use the form `Sin(θ/12)^5`, not `Sin^5[θ/12]`, which Mathematica will not interpret in the way intended.

```
Clear[r]
r[θ_] = Exp[Cos[θ]] - 2Cos[4θ] + Sin[θ/12]^5;
pp3 = PolarPlot[r[θ], {θ, 0, 24π}, PlotPoints → 200,
  PlotRange → {{-4, 5}, {-4.5, 4.5}},
  AspectRatio → Automatic];
Clear[r]
pp4 = PolarPlot[{Sqrt[1/θ], -Sqrt[1/θ]}, {θ, .1, 10π},
  AspectRatio → Automatic, PlotRange → All];
```

Finally, we use `Show` together with `GraphicsGrid` to display all four graphs as a graphics array in Figure 2.16. `pp1` and `pp2` are shown in the first row and `pp3` and `pp4` are shown in the second.

```
Show[GraphicsGrid[{{pp1, pp2}, {pp3, pp4}}]]
```

**FIGURE 2.16**

Graphs of four polar equations

2.3.3 Three-Dimensional and Contour Plots: Graphing Equations

An elementary function of two variables, $z = f(x, y) = \text{expression in } x \text{ and } y$, is typically defined using the form

$$f[x_, y_] = \text{expression in } x \text{ and } y.$$

For delayed evaluation, use `f[x_, y_] := ...` rather than `f[x_, y_] = ...` (immediate evaluation). Once a function has been defined, a basic graph is generated with `Plot3D`:

$$\text{Plot3D}[f[x, y], \{x, a, b\}, \{y, c, d\}]$$

graphs $f(x, y)$ for $a \leq x \leq b$ and $c \leq y \leq d$.

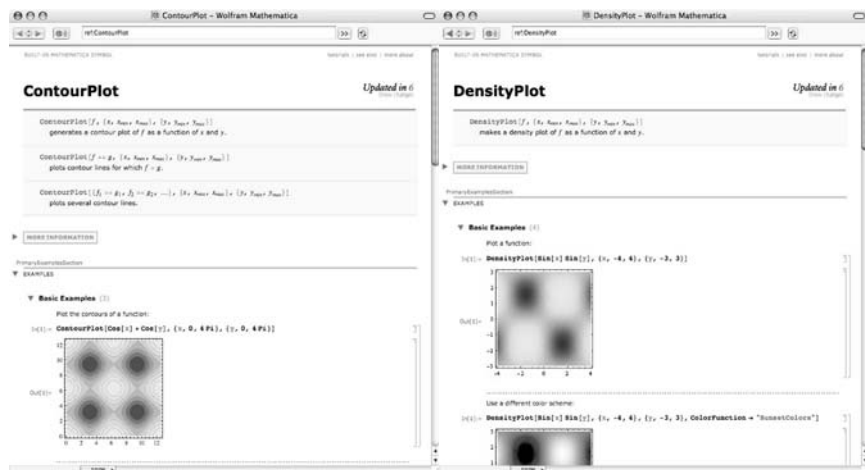
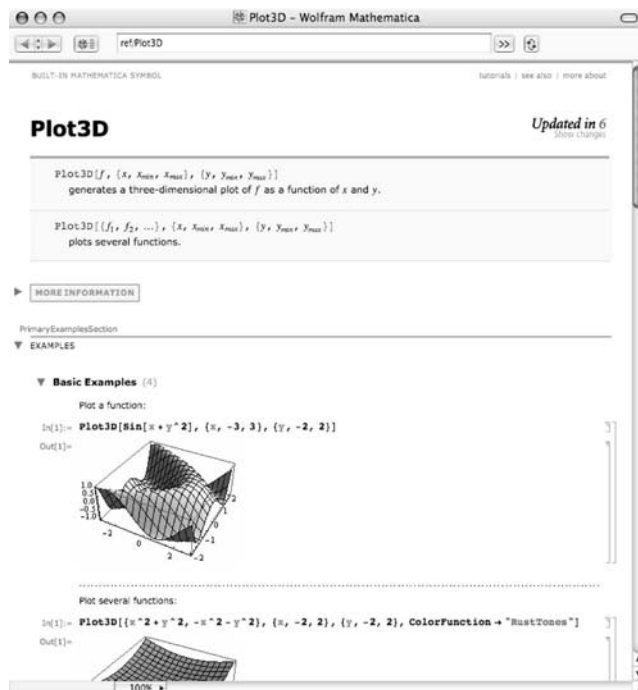
For details regarding `Plot3D` and its options, enter `?Plot3D` or `??Plot3D` or access the **Documentation Center** to obtain information about the `Plot3D` command, as we do here.

Graphs of several level curves of $z = f(x, y)$ are generated with

$$\text{ContourPlot}[f[x, y], \{x, a, b\}, \{y, c, d\}].$$

A density plot of $z = f(x, y)$ is generated with

$$\text{DensityPlot}[f[x, y], \{x, a, b\}, \{y, c, d\}].$$



For details regarding ContourPlot (DensityPlot) and its options, enter ?ContourPlot (?DensityPlot) or ??ContourPlot (??DensityPlot) or access the **Documentation Center**.

Example 2.3.12 Let $f(x, y) = \frac{x^2 y}{x^4 + 4y^2}$. (a) Calculate $f(1, -1)$. (b) Graph $f(x, y)$ and several contour plots of $f(x, y)$ on a region containing $(0, 0)$.

Solution After defining $f(x, y)$, we evaluate $f(1, -1) = -1/5$.

```
Clear[f]
f[x_, y_] = x^2 y / (x^4 + 4 y^2)

$$\frac{x^2 y}{x^4 + 4 y^2}$$

f[1, 1]

$$-\frac{1}{5}$$

```

Next, we use `Plot3D` to graph $f(x, y)$ for $-1/2 \leq x \leq 1/2$ and $-1/2 \leq y \leq 1/2$ in Figure 2.17. We illustrate the use of the `Axes`, `Boxed`, `PlotPoints`, `MeshFunctions`, `PlotStyle`, and `ColorFunction` options.

```
p1 = Plot3D[f[x, y], {x, -1/2, 1/2}, {y, -1/2, 1/2},
  Axes → Automatic, Boxed → False, PlotPoints → 50]
```

Use `MeshFunctions` to modify the standard rectangular grid. In Figure 2.17(b), we use the level curves of the function for the grid.

```
p2 = Plot3D[f[x, y], {x, -1/2, 1/2}, {y, -1/2, 1/2},
  Axes → Automatic, Boxed → False, MeshFunctions → {#3 &},
  PlotPoints → 50]
```

We use the `GrayTones` color gradient to shade the graph (Figure 2.17(c))

```
p3 = Plot3D[f[x, y], {x, -1/2, 1/2}, {y, -1/2, 1/2},
  Axes → Automatic, Boxed → False, MeshFunctions → {#3 &},
  PlotPoints → 50, ColorFunction → (ColorData["GrayTones"][#3] &)]
```

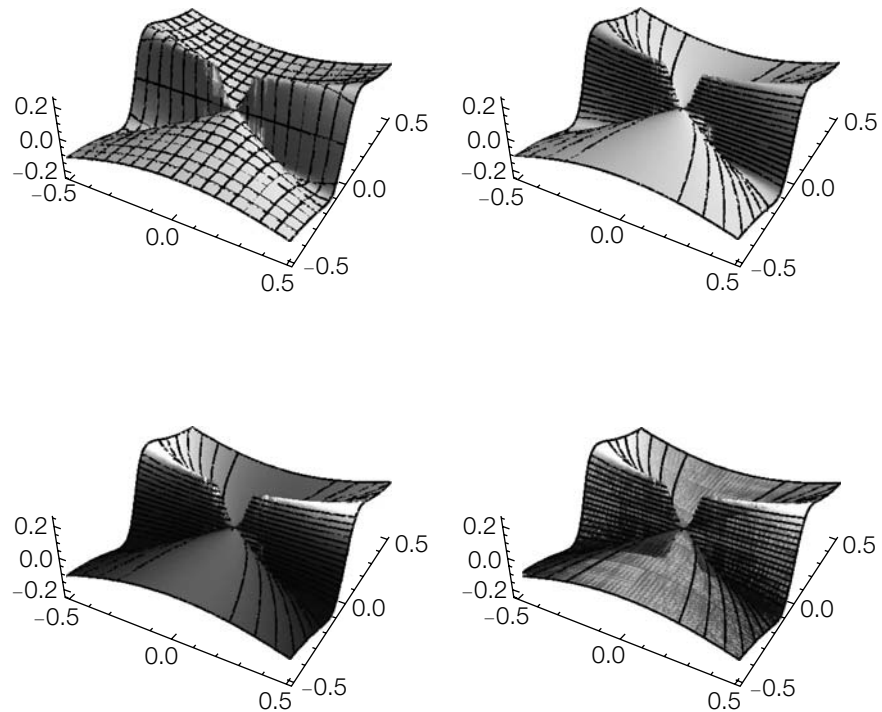
Use `Opacity` to make a “clear” plot (Figure 2.17(d)). We use `Show` together with `GraphicsGrid` to display all four plots together in Figure 2.17.

```
p4 = Plot3D[f[x, y], {x, -1/2, 1/2}, {y, -1/2, 1/2},
  Axes → Automatic, Boxed → False, MeshFunctions → {#3 &},
  PlotPoints → 50, ColorFunction → (ColorData["GrayTones"][#3] &),
  PlotStyle → Opacity[.3]]
Show[GraphicsGrid[{{p1, p2}, {p3, p4}}]]
```

Four contour plots are generated with `ContourPlot`. The second through fourth illustrate the use of the `PlotPoints`, `Frame`, `ContourShading`, `Axes`, `AxesOrigin`, `ColorFunction`, and `Contours` options (see Figure 2.18).

```
cp1 = ContourPlot[f[x, y], {x, -1/2, 1/2}, {y, -1/2, 1/2},
  PlotPoints → 50]
cp2 = ContourPlot[f[x, y], {x, -1/2, 1/2}, {y, -1/2, 1/2},
```

To adjust the viewing angle of three-dimensional graphics, select the graphic and drag to the desired viewing angle.

**FIGURE 2.17**

Three-dimensional plot of $f(x,y)$: Upper left is the basic plot generated with `Plot3D`; in upper right, we use contour lines to determine the mesh; in lower left, we use the `GrayTones` color gradient to shade the plot; in lower right, we create a transparent plot with `Opacity`

```

Axes → Automatic, PlotPoints → 50,
ColorFunction → ColorData["GrayTones"]]

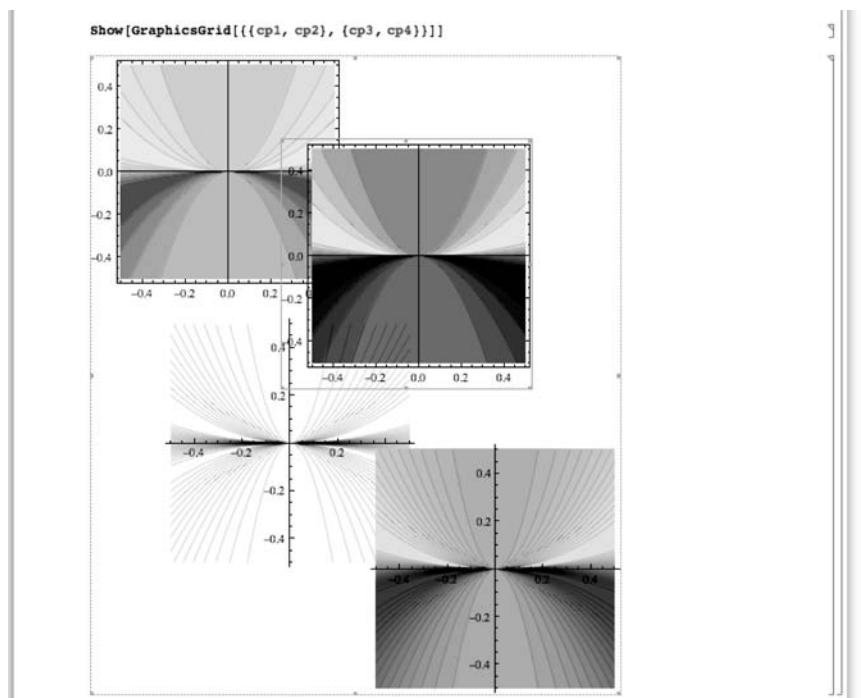
cp3 = ContourPlot[f[x, y], {x, -1/2, 1/2}, {y, -1/2, 1/2},
Axes → Automatic, PlotPoints → 50, Contours → 30,
ContourShading → False, Frame → False,
Axes → Automatic, AxesOrigin → {0, 0}]

cp4 = ContourPlot[f[x, y], {x, -1/2, 1/2}, {y, -1/2, 1/2},
Axes → Automatic, PlotPoints → 50, Contours → 30,
Frame → False, ColorFunction → "CandyColors",
Axes → Automatic, AxesOrigin → {0, 0}]

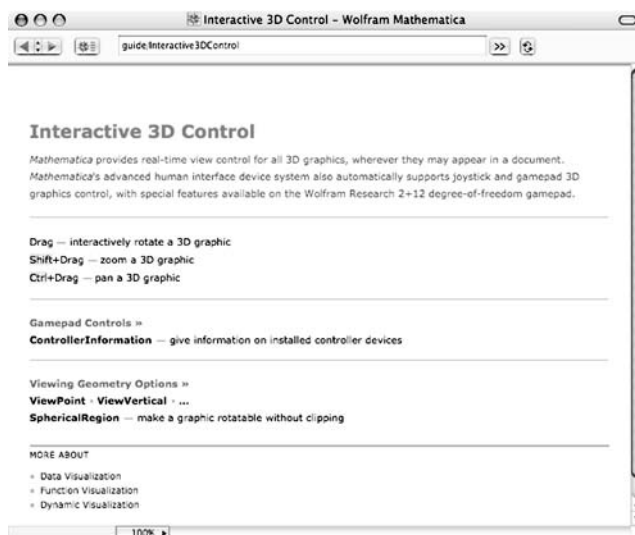
Show[GraphicsGrid[{{cp1, cp2}, {cp3, cp4}}]]

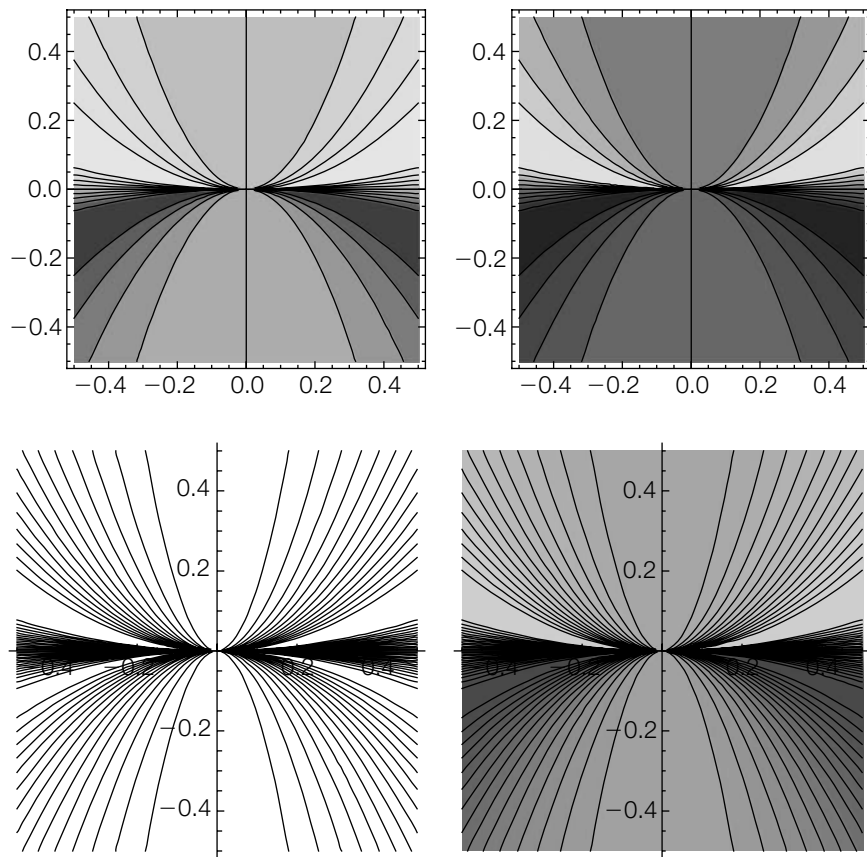
```

Figure 2.18 shows the graphics array generated with the previous commands. With Mathematica 6, if you want to adjust your array, drag and move the objects within the graphic.



With Mathematica 6, you can adjust the viewing angle of a 3D graphic by selecting the graphic and dragging it to the desired position. Manually, use the ViewPoint option.

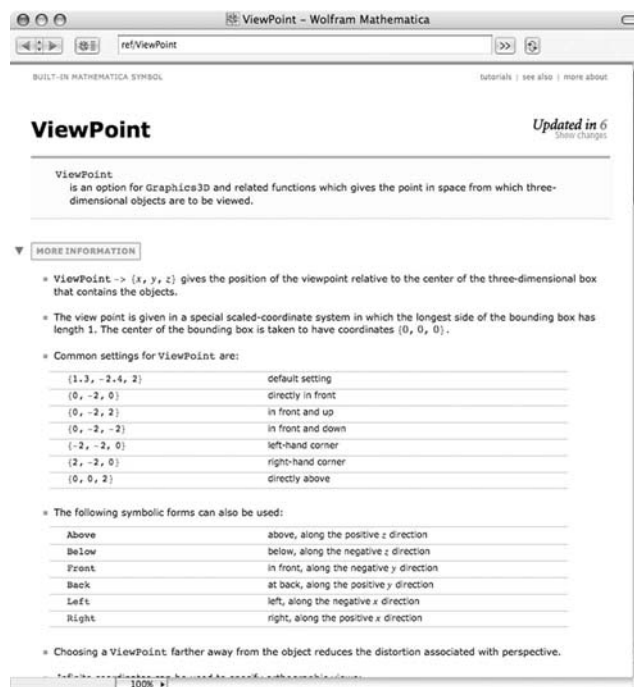


**FIGURE 2.18**

Four contour plots of $f(x,y)$: upper left, the basic plot generated by `ContourPlot`; upper right, introduce a coloring function; lower left, eliminate the coloring function and increase the number of contours; lower right, add color and increase the number of contours

Figure 2.19 shows four different views of the graph of $g(x,y) = x \sin y + y \sin x$ for $0 \leq x \leq 5\pi$ and $0 \leq y \leq 5\pi$. The options `AxesLabel`, `BoxRatios`, `ViewPoint`, `PlotPoints`, `Shading`, and `Mesh` are also illustrated.

```
Clear[g]
g[x_, y_] = x Sin[y] + y Sin[x];
p1 = Plot3D[g[x, y], {x, 0, 5Pi}, {y, 0, 5Pi},
  PlotPoints -> 60, AxesLabel -> {"x", "y", "z"}];
p2 = Plot3D[g[x, y], {x, 0, 5Pi}, {y, 0, 5Pi},
  PlotPoints -> 60, ViewPoint -> {-2.846, -1.813, 0.245},
  Boxed -> False, BoxRatios -> {1, 1, 1},
  AxesLabel -> {"x", "y", "z"}];
```



```

p3 = Plot3D[g[x, y], {x, 0, 5π}, {y, 0, 5π},
  PlotPoints → 60, ViewPoint → {1.488, -1.515, 2.634},
  AxesLabel → {"x", "y", "z"}, ColorFunction → (White&)];
p4 = Plot3D[g[x, y], {x, 0, 5Pi}, {y, 0, 5Pi},
  PlotPoints → 60, AxesLabel → {"x", "y", "z"},
  Mesh → False, BoxRatios → {2, 2, 3},
  ViewPoint → {-1.736, 1.773, -2.301}];
Show[GraphicsGrid[{{p1, p2}, {p3, p4}}]]

```

ContourPlot is especially useful when graphing equations. The graph of the equation $f(x, y) = C$, where C is a constant, is the same as the contour plot of $z = f(x, y)$ corresponding to C . That is, the graph of $f(x, y) = C$ is the same as the level curve of $z = f(x, y)$ corresponding to $z = C$.

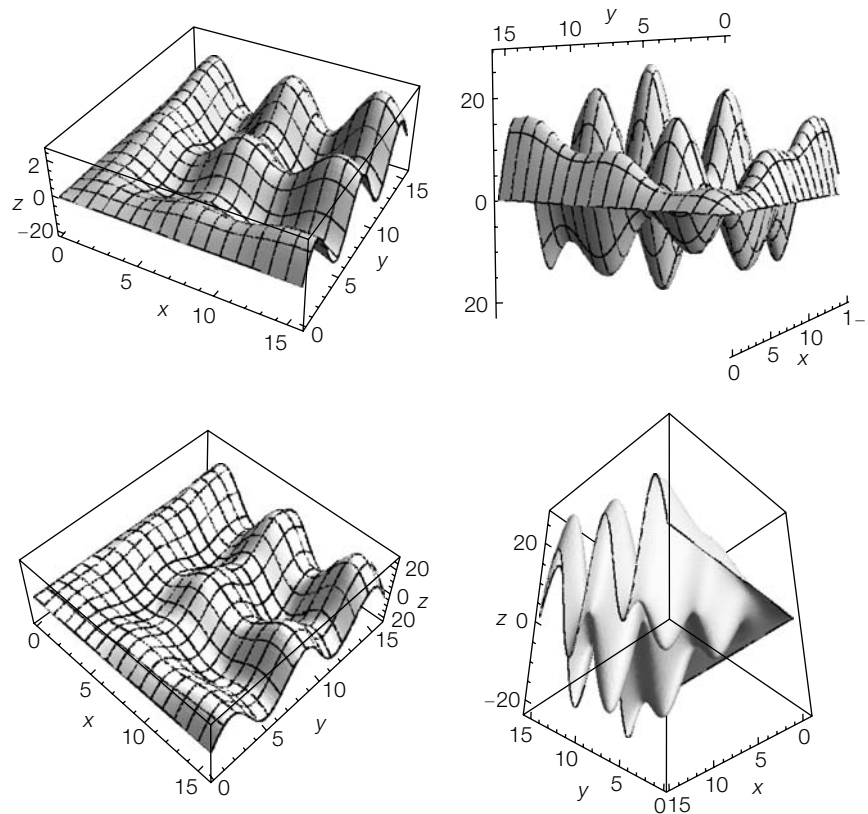
Example 2.3.13 Graph the unit circle, $x^2 + y^2 = 1$.

Solution We first graph $z = x^2 + y^2$ for $-4 \leq x \leq 4$ and $-4 \leq y \leq 4$ with Plot3D in Figure 2.20(a).

```

p1 = Plot[{Sqrt[1 - x^2], -Sqrt[1 - x^2]}, {x, -1, 1},
  PlotRange → {{-3/2, 3/2}, {-3/2, 3/2}},
  AspectRatio → Automatic];

```

**FIGURE 2.19**

Four different plots of $g(x, y) = x \sin y + y \sin x$ for $0 \leq x \leq 5\pi$

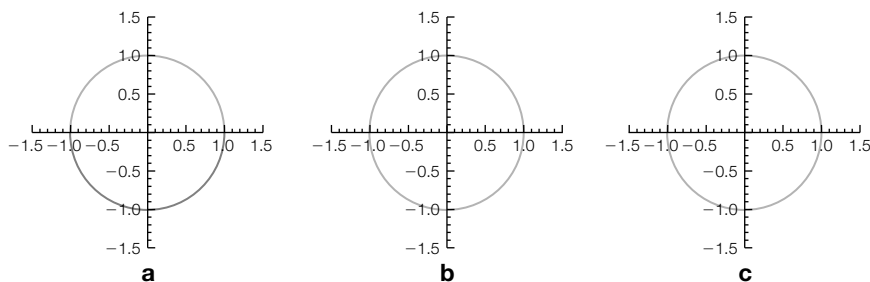
The graph of $x^2 + y^2 = 1$ is the graph of $z = x^2 + y^2$ corresponding to $z = 1$ as well as the graph of $(\cos t, \sin t)$ for $0 \leq t \leq 2\pi$. We use `ParametricPlot` to graph these parametric equations in Figure 2.20.

```
p2 = ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2Pi},
  PlotRange → {{-3/2, 3/2}, {-3/2, 3/2}},
  AspectRatio → Automatic];
```

For the unit circle, it is probably easiest to convert to polar coordinates and use `PolarPlot`.

```
p3 = PolarPlot[1, {t, 0, 2Pi},
  PlotRange → {{-3/2, 3/2}, {-3/2, 3/2}},
  AspectRatio → Automatic];
Show[GraphicsRow[{p1, p2, p3}]]
```

When converting from rectangular to polar coordinates, use $x = r \cos \theta$ and $y = r \sin \theta$.

**FIGURE 2.20**

Three different ways of generating plots of the unit circle—all plots are the same in the end

Use `ContourPlot` to graph equations of the form $f(x,y) = g(x,y)$ with `ContourPlot[f[x,y]==g[x,y],{x,a,b},{y,c,d}]`.

Example 2.3.14 Graph the equation $y^2 - 2x^4 + 2x^6 - x^8 = 0$ for $-1.5 \leq x \leq 1.5$.

Solution We define `lhseq` to be the left-hand side of the equation $y^2 - 2x^4 + 2x^6 - x^8 = 0$ and then use `ContourPlot` to graph `eq` for $-1.5 \leq x \leq 1.5$ in Figure 2.21.

```
Clear[x,y]
lhseq = y^2 - x^4 + 2x^6 - x^8;
cp1 = ContourPlot[lhseq==0,{x,-2,2},{y,-2,2},
  AspectRatio -> Automatic]
cp2 = ContourPlot[lhseq==0,{x,-2,2},{y,-2,2},
  AspectRatio -> Automatic, Frame -> False,
  Axes -> Automatic, AxesLabel -> {x,y}]
Show[GraphicsRow[{cp1, cp2}]]
```

Equations can be plotted together, as with the commands `Plot` and `Plot3D`, with

`ContourPlot[{eq1, eq2, ..., eqn},{x,a,b},{y,c,d}]`.

Example 2.3.15 Graph the equations $x^2 + y^2 = 1$ and $4x^2 - y^2 = 1$ for $-1.5 \leq x \leq 1.5$.

Solution We use `ContourPlot` to graph the equations together on the same axes in Figure 2.22. The graph of $x^2 + y^2 = 1$ is the unit circle, whereas the graph of $4x^2 - y^2 = 1$ is a hyperbola.

```
cp1 = ContourPlot[{x^2 + y^2==1, 4x^2 - y^2==1},
  {x,-3/2,3/2},{y,-3/2,3/2}, Frame -> False,
  Axes -> Automatic]
```

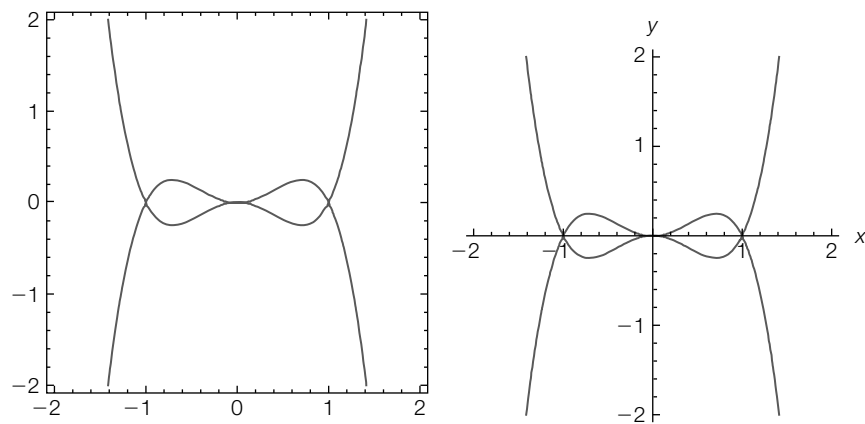


FIGURE 2.21

Two plots of $y^2 - 2x^4 + 2x^6 - x^8 = 0$

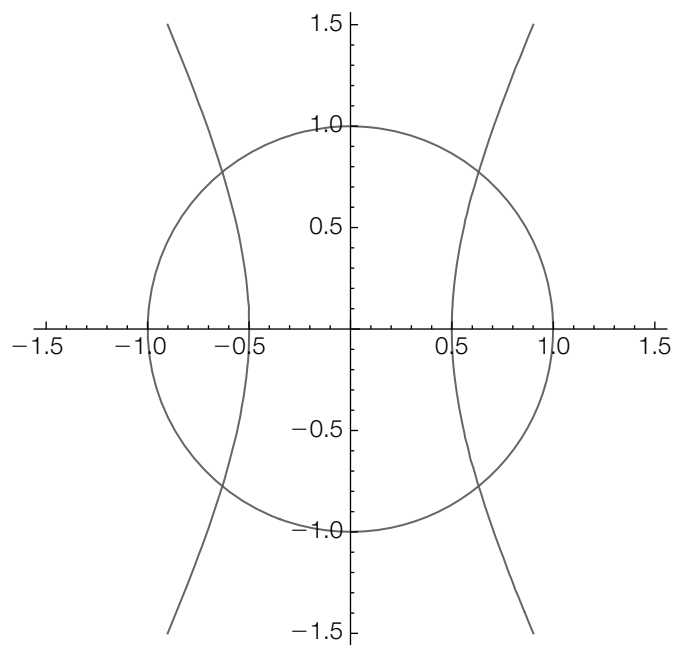


FIGURE 2.22

Plots of $x^2 + y^2 = 1$ and $4x^2 - y^2 = 1$

Example 2.3.16 (Conic Sections). A **conic section** is a graph of the equation

Also see Example 2.3.19.

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0.$$

Except when the conic is degenerate, the conic $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$ is a (an)

1. **Ellipse** or **circle** if $B^2 - 4AC < 0$;
2. **Parabola** if $B^2 - 4AC = 0$; or
3. **Hyperbola** if $B^2 - 4AC > 0$.

Graph the conic section $ax^2 + bxy + cy^2 = 1$ for $-4 \leq x \leq 4$ and for a , b , and c equal to all possible combinations of -1 , 1 , and 2 .

Solution We begin by defining `conic` to be the equation $ax^2 + bxy + cy^2 = 1$ and then use `Permutations` to produce all possible orderings of the list of numbers $\{-1, 1, 2\}$, naming the resulting output `vals`.

`Permutations[list]` returns a list of all possible orderings of the list `list`.

```
Clear[a, b, c, x, y, p]
conic = ax^2 + bxy + cy^2 == 1;
vals = Permutations[{-1, 1, 2}]
{{-1, 1, 2}, {-1, 2, 1}, {1, -1, 2},
 {1, 2, -1}, {2, -1, 1}, {2, 1, -1}}
```

Next we define the function `p`. Given `a1`, `b1`, and `c1`, `p` defines `toplot` to be the equation obtained by replacing a , b , and c in `conic` by `a1`, `b1`, and `c1`, respectively. Then, `toplot` is graphed for $-4 \leq x \leq 4$. `p` returns a graphics object.

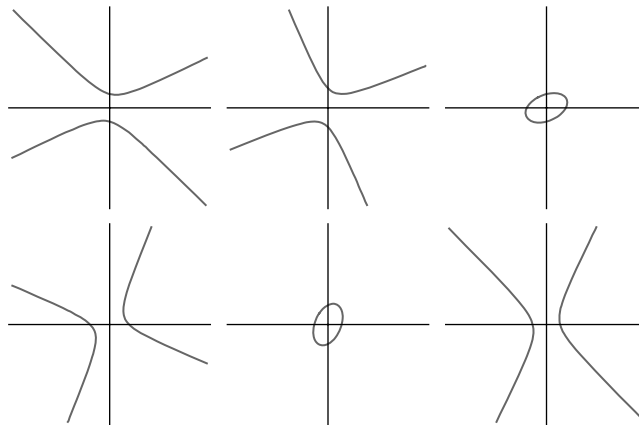
```
Clear[p]
p[{a1_, b1_, c1_}] := Module[{toplot},
  toplot = Evaluate[conic /. {a -> a1, b -> b1, c -> c1}];
  ContourPlot[Evaluate[toplot],
    {x, -5, 5}, {y, -5, 5}, Frame -> False,
    Axes -> Automatic, Ticks -> None]
]
```

We then use `Map` to compute `p` for each ordered triple in `vals`. The resulting output, named `graphs`, is a set of six graphics objects.

```
graphs = Map[p, vals];
```

`Partition` is then used to partition `graphs` into three element subsets. The resulting array of graphics objects named `toshow` is displayed with `Show` and `GraphicsGrid` in Figure 2.23.

```
Show[GraphicsGrid[Partition[graphs, 3]]]
```

**FIGURE 2.23**

Plots of six conic sections

2.3.4 Parametric Curves and Surfaces in Space

The command

ParametricPlot3D[{x[t], y[t], z[t]}, {t, a, b}]

generates the three-dimensional curve $\begin{cases} x = x(t), \\ y = y(t), \\ z = z(t), \end{cases} \quad a \leq t \leq b$, and the

command

ParametricPlot3D[{x[u, v], y[u, v], z[u, v]}, {u, a, b}, {v, c, d}]

plots the surface $\begin{cases} x = x(u, v), \\ y = y(u, v), \\ z = z(u, v), \end{cases} \quad a \leq u \leq b, c \leq v \leq d.$

Entering `Information[ParametricPlot3D]` or `??ParametricPlot3D` returns a description of the `ParametricPlot3D` command along with a list of options and their current settings.

Example 2.3.17 (Umbilic Torus NC). A parametrization of **umbilic torus NC** is given by $\mathbf{r}(s, t) = x(s, t)\mathbf{i} + y(s, t)\mathbf{j} + z(s, t)\mathbf{k}$, $-\pi \leq s \leq \pi$, $-\pi \leq t \leq \pi$, where

$$\begin{aligned} x &= \left[7 + \cos\left(\frac{1}{3}s - 2t\right) + 2\cos\left(\frac{1}{3}s + t\right) \right] \sin s \\ y &= \left[7 + \cos\left(\frac{1}{3}s - 2t\right) + 2\cos\left(\frac{1}{3}s + t\right) \right] \cos s \end{aligned}$$

and

$$z = \sin\left(\frac{1}{3}s - 2t\right) + 2\sin\left(\frac{1}{3}s + t\right).$$

Graph the torus.

Solution We define x , y , and z .

```
c = 3;
a = 1;
x[s_, t_] = (7 + Cos[s/3 - 2t] + 2Cos[s/3 + t])Sin[s];
y[s_, t_] = (7 + Cos[s/3 - 2t] + 2Cos[s/3 + t])Cos[s];
z[s_, t_] = Sin[s/3 - 2t] + 2Sin[s/3 + t];
r[s_, t_] = {x[s, t], y[s, t], z[s, t];}
```

The torus is then graphed with `ParametricPlot3D`, `DensityPlot`, and `ContourPlot` in Figure 2.24. In the plots, we illustrate the `Mesh`, `MeshFunctions`, `PlotPoints`, and `PlotRange` options. All four plots are shown together with `Show` and `GraphicsGrid`. Notice that `DensityPlot` and `ContourPlot` yield very similar results: A basic density plot is like a basic contour plot without the contour lines.

```
threedp1uta = ParametricPlot3D[r[s, t], {s, -Pi, Pi},
  {t, -Pi, Pi}, PlotPoints -> {30, 30},
  AspectRatio -> 1, AxesLabel -> {"x", "y", "z"},
  PlotRange -> {{-12, 12}, {-12, 12}, {-3, 3}},
  BoxRatios -> {4, 4, 1}, Mesh -> False, PlotStyle -> Opacity[.9]]
threedp1utb = ParametricPlot3D[r[s, t], {s, -Pi, Pi},
  {t, -Pi, Pi}, PlotPoints -> {50, 50},
  AspectRatio -> 1, AxesLabel -> {"x", "y", "z"},
  PlotRange -> {{-12, 12}, {-12, 12}, {-3, 3}},
  BoxRatios -> {4, 4, 1},
  MeshFunctions -> {#3&}, Mesh -> 10]
threedp1utc = DensityPlot[r[s, t], {s, -Pi, Pi},
  {t, -Pi, Pi}, PlotPoints -> {100, 100},
  AspectRatio -> 1,
  MeshFunctions -> {#3&}, Mesh -> 10]
threedp1utd = ContourPlot[r[s, t], {s, -Pi, Pi},
  {t, -Pi, Pi}, PlotPoints -> {100, 100},
  AspectRatio -> 1,
  MeshFunctions -> {#3&}, Mesh -> 10]
Show[GraphicsGrid[{{threedp1uta, threedp1utb},
  {threedp1utc, threedp1utd}}]]
```

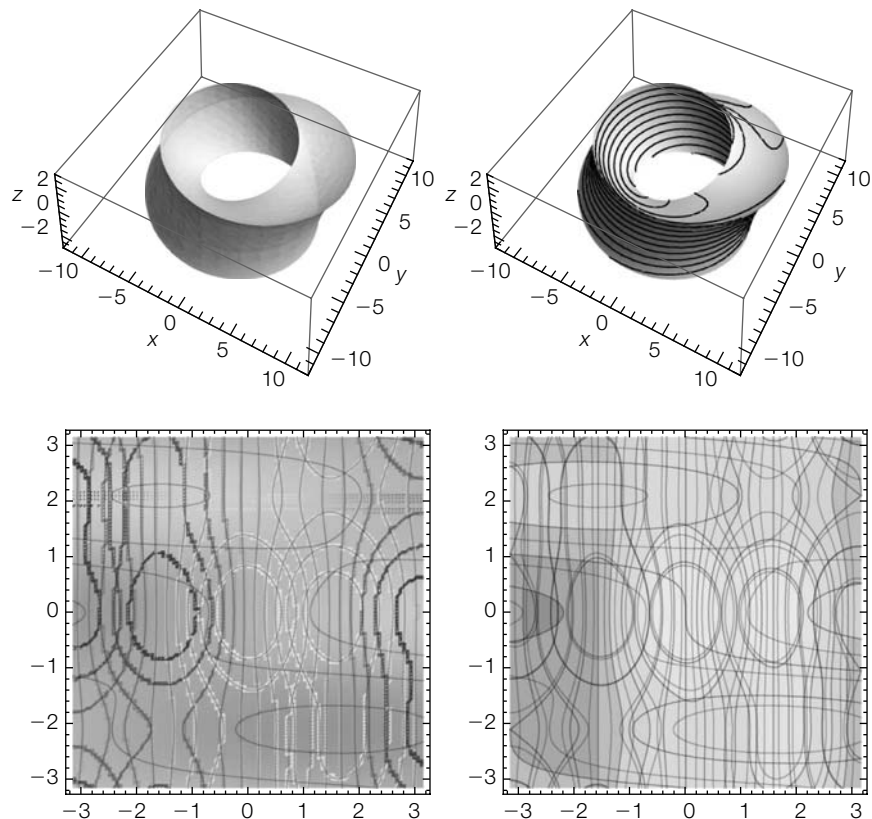


FIGURE 2.24

On the top row, two plots of umbilic torus; on the bottom, comparing a density plot (left) to a contour plot (right)

Example 2.3.18 (Gray's Torus Example). A parametrization of an **elliptical torus** is given by

This example is explored in detail in Sections 8.2 and 11.4 of Gray's *Modern Differential Geometry of Curves and Surfaces* [11], an indispensable reference for those who use Mathematica's graphics extensively.

$$x = (a + b \cos v) \cos u, \quad y = (a + b \cos v) \sin u, \quad z = c \sin v$$

For positive integers p and q , the curve with parametrization

$$x = (a + b \cos qt) \cos pt, \quad y = (a + b \cos qt) \sin pt, \quad z = c \sin qt$$

winds around the elliptical torus and is called a **torus knot**.

Plot the torus if $a = 8$, $b = 3$, and $c = 5$ and then graph the torus knots for $p = 2$ and $q = 5$, $p = 1$ and $q = 10$, and $p = 2$ and $q = 3$.

Solution We begin by defining torus and torusknot.

```
torus[a_, b_, c_][p_, q_][u_, v_] :=
  {(a + b Cos[u])Cos[v], (a + b Cos[u])Sin[v], c Sin[u]}
torusknot[a_, b_, c_][p_, q_][t_] :=
  {(a + b Cos[qt])Cos[pt], (a + b Cos[qt])Sin[pt], c Sin[qt]}
```

Next, we use ParametricPlot3D to generate all four graphs

```
pp1 = ParametricPlot3D[Evaluate[torus[8, 3, 5][2, 5][u, v],
  {u, 0, 2Pi}, {v, 0, 2Pi}], PlotPoints -> 60];
pp2 = ParametricPlot3D[Evaluate[torusknot[8, 3, 5][2, 5][t],
  {t, 0, 3Pi}], PlotPoints -> 200];
pp3 = ParametricPlot3D[Evaluate[torusknot[8, 3, 5][1, 10][t],
  {t, 0, 3Pi}], PlotPoints -> 200];
pp4 = ParametricPlot3D[Evaluate[torusknot[8, 3, 5][2, 3][t],
  {t, 0, 3Pi}], PlotPoints -> 200];
```

and show the result as a graphics array with Show and GraphicsGrid in Figure 2.25.

```
Show[GraphicsGrid[{{pp1, pp2}, {pp3, pp4}}]]
```

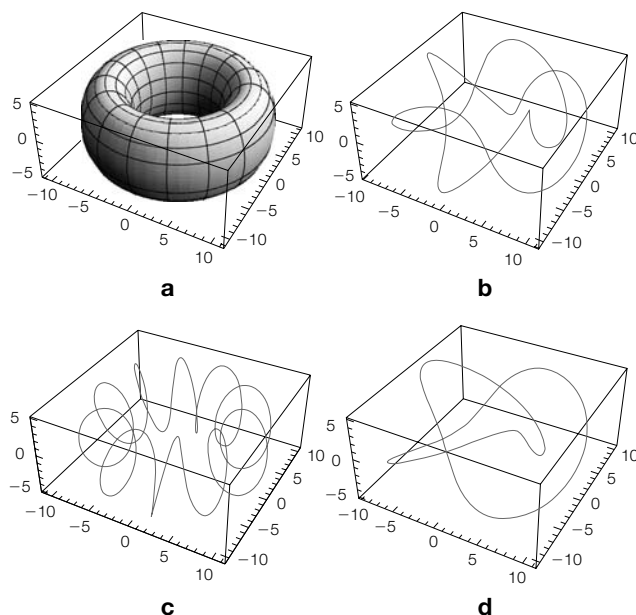


FIGURE 2.25

(a) An elliptical torus. (b) This knot is also known as the trefoil knot. (c) The curve generated by `torusknot[8,3,5][2,3][1,10]` is not a knot. (d) The torus knot with $p = 2$ and $q = 3$

If we take advantage of a few options, such as eliminating the mesh (**Mesh**->**False**) and increasing the opacity (**PlotStyle**->**Opacity** [.4]), we can produce a graphic of the knot on the torus. After using the **PlotStyle** option together with **Opacity**, we produce a nearly transparent torus. Then, each knot is plotted. To ensure smooth plots, we increase the number of points plotted with **PlotPoints** and also increase the thickness of the curve with **Thickness**.

```
pp1 = ParametricPlot3D[Evaluate[torus[8, 3, 5][2, 5][u, v]],
  {u, 0, 2Pi}, {v, 0, 2Pi}, PlotPoints -> 60,
  Mesh -> False, PlotStyle -> Opacity[.4],
  ColorFunction -> "AlpineColors"];
pp2 = ParametricPlot3D[Evaluate[torusknot[8, 3, 5][2, 5][t]],
  {t, 0, 3Pi}, PlotPoints -> 200, PlotStyle -> {{Thickness[.01]}}];
pp3 = ParametricPlot3D[Evaluate[torusknot[8, 3, 5][1, 10][t]],
  {t, 0, 3Pi}, PlotPoints -> 200, PlotStyle -> {{Thickness[.01]}}];
pp4 = ParametricPlot3D[Evaluate[torusknot[8, 3, 5][2, 3][t]],
  {t, 0, 3Pi}, PlotPoints -> 200, PlotStyle -> {{Thickness[.01]}}];
```

We use **Show** twice together with **GraphicsRow** to first display the torus with each knot and then display all three graphics side-by-side in Figure 2.26.

```
Show[GraphicsRow[{Show[{pp1, pp2}], Show[{pp1, pp3}], Show[{pp1, pp4}]}]]
```

Example 2.3.19 (Quadric Surfaces). The **quadric surfaces** are the three-dimensional objects corresponding to the conic sections in two dimensions. A **quadric surface** is a graph of

Also see Example 2.3.16.

$$Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0,$$

where A – J are constants.

The intersection of a plane and a quadric surface is a conic section.

Several of the basic quadric surfaces, in standard form, and a parametrization of the surface are listed in the following table.

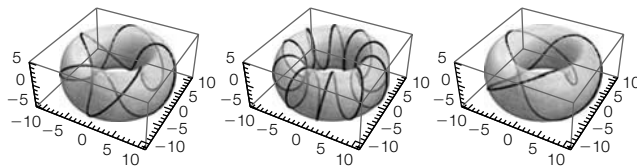


FIGURE 2.26

The knots in Figure 2.25 on the torus

Name	Parametric Equations
Ellipsoid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \quad \begin{cases} x = a \cos t \cos r, \\ y = b \cos t \sin r, \\ z = c \sin t, \end{cases} \quad -\pi/2 \leq t \leq \pi/2, -\pi \leq r \leq \pi$
Hyperboloid of One Sheet	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1 \quad \begin{cases} x = a \sec t \cos r, \\ y = b \sec t \sin r, \\ z = c \tan t, \end{cases} \quad -\pi/2 < t < \pi/2, -\pi \leq r \leq \pi$
Hyperboloid of Two Sheets	$\frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1 \quad \begin{cases} x = a \sec t, \\ y = b \tan t \cos r, \\ z = c \tan t \sin r, \end{cases} \quad \begin{array}{l} -\pi/2 < t < \pi/2 \text{ or} \\ \pi/2 < t < 3\pi/2, -\pi \leq r \leq \pi \end{array}$

Graph the ellipsoid with equation $\frac{1}{16}x^2 + \frac{1}{4}y^2 + z^2 = 1$, the hyperboloid of one sheet with equation $\frac{1}{16}x^2 + \frac{1}{4}y^2 - z^2 = 1$, and the hyperboloid of two sheets with equation $\frac{1}{16}x^2 - \frac{1}{4}y^2 - z^2 = 1$.

Solution A parametrization of the ellipsoid with equation $\frac{1}{16}x^2 + \frac{1}{4}y^2 + z^2 = 1$ is given by

$$x = 4 \cos t \cos r, \quad y = 2 \cos t \sin r, \quad z = \sin t, \quad -\pi/2 \leq t \leq \pi/2, -\pi \leq r \leq \pi,$$

which is graphed with ParametricPlot3D.

```
Clear[x, y, z]
x[t_, r_] = 4Cos[t]Cos[r];
y[t_, r_] = 2Cos[t]Sin[r];
z[t_, r_] = Sin[t];
pp1 = ParametricPlot3D[{x[t, r], y[t, r], z[t, r]}, {t, -Pi/2, Pi/2},
  {r, -Pi, Pi}, PlotPoints -> 30];
```

A parametrization of the hyperboloid of one sheet with equation $\frac{1}{16}x^2 + \frac{1}{4}y^2 - z^2 = 1$ is given by

$$x = 4 \sec t \cos r, \quad y = 2 \sec t \sin r, \quad z = \tan t, \quad -\pi/2 < t < \pi/2, -\pi \leq r \leq \pi.$$

Because $\sec t$ and $\tan t$ are undefined if $t = \pm\pi/2$, we use ParametricPlot3D to graph these parametric equations on a subinterval of $[-\pi/2, \pi/2]$, $[-\pi/3, \pi/3]$.

```
Clear[x, y, z]
x[t_, r_] = 4Sec[t]Cos[r];
```

```

y[t_, r_] = 2Sec[t]Sin[r];
z[t_, r_] = Tan[t];
pp2 = ParametricPlot3D[{x[t, r], y[t, r], z[t, r]}, {t, -Pi/3, Pi/3},
  {r, -Pi, Pi}, PlotPoints -> 30];

```

pp1 and pp2 are shown together in Figure 2.27 using Show and GraphicsRow.

```
Show[GraphicsRow[{pp1, pp2}]]
```

For (c), we take advantage of the ContourPlot3D function:

```
ContourPlot3D[f[x, y, z], {x, a, b}, {y, c, d}, {z, u, v}]
```

graphs several level surfaces of $w = f(x, y, z)$.

We use ContourPlot3D to graph the equation $\frac{1}{16}x^2 - \frac{1}{4}y^2 - z^2 - 1 = 0$ in Figure 2.28(a), illustrating the use of the PlotPoints, Axes, AxesLabel, and BoxRatios options. In Figure 2.28(b), several level surfaces are drawn that illustrate the use of the Opacity function with the ContourStyle and Mesh options.

```

cp3d1 = ContourPlot3D[x^2/16 - y^2/4 - z^2 - 1 == 0,
  {x, -10, 10}, {y, -8, 8}, {z, -2, 2},
  PlotPoints -> {8, 8, 8}, Axes -> Automatic,
  AxesLabel -> {"x", "y", "z"}, BoxRatios -> {2, 1, 1}]

```

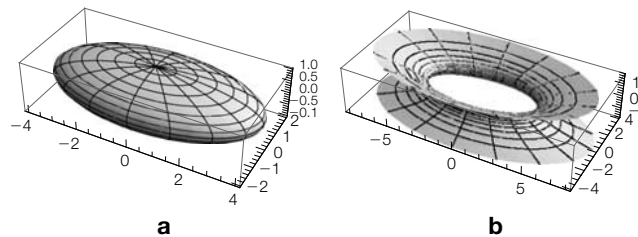


FIGURE 2.27

(a) Plot of $\frac{1}{16}x^2 + \frac{1}{4}y^2 + z^2 = 1$. (b) Plot of $\frac{1}{16}x^2 + \frac{1}{4}y^2 - z^2 = 1$

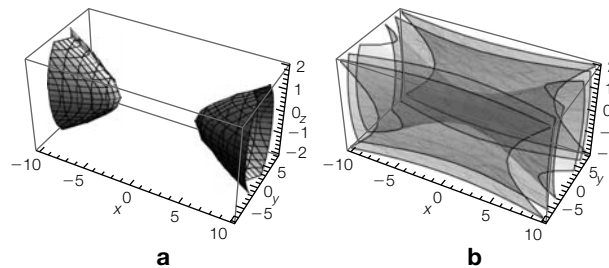


FIGURE 2.28

(a) Plot of $\frac{1}{16}x^2 - \frac{1}{4}y^2 - z^2 = 1$ generated with ContourPlot3D. (b) Several level surfaces of $f(x, y, z) = \frac{1}{16}x^2 - \frac{1}{4}y^2 - z^2$

```

cp3d2 = ContourPlot3D[x^2/16 - y^2/4 - z^2 - 1,
  {x, -10, 10}, {y, -8, 8},
  {z, -2, 2}, PlotPoints → {8, 8, 8}, Axes → Automatic,
  AxesLabel → {"x", "y", "z"},
  BoxRatios → {2, 1, 1}, Mesh → False,
  ContourStyle → Opacity[.5]]
Show[GraphicsRow[{cp3d1, cp3d2}]]

```

ContourPlot3D is especially useful in plotting equations involving three variables x , y , and z for which it is difficult to solve for one variable as a function of the other two.

Example 2.3.20 (Cross-Cap). The **Cross-Cap** has equation

$$4x^2(x^2 + y^2 + z^2 + z) + y^2(y^2 + z^2 - 1) = 0.$$

We use ContourPlot3D to generate the plot of the cross-cap shown in Figure 2.29.

```

ContourPlot3D[4x^2(x^2 + y^2 + z^2 + z) +
  y^2(y^2 + z^2 - 1) == 0, {x, -1, 1}, {y, -1, 1},
  {z, -1, 1}, Mesh → False, Boxed → False,
  Axes → None, ColorFunction → (ColorData["BrightBands"][#3]&),
  ContourStyle → Opacity[.8]]

```

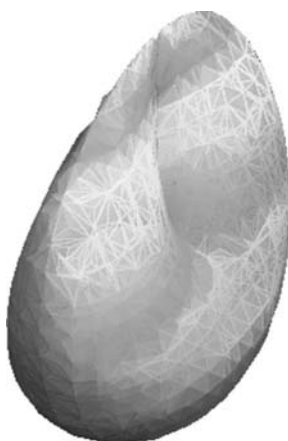


FIGURE 2.29

The Cross-Cap

Example 2.3.21 A homotopy from the **Roman surface** to the **Boy surface** is given by

If f and g are functions from X to Y , a **homotopy** from f to g is a continuous function H from $X \times [0, 1]$ to Y satisfying $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$.

$$\begin{aligned}x(u, v) &= \frac{\sqrt{2} \cos(2u) \cos^2 v + \cos u \sin(2v)}{2 - \alpha \sqrt{2} \sin(3u) \sin(2v)}, \\y(u, v) &= \frac{\sqrt{2} \sin(2u) \cos^2 v + \sin u \sin(2v)}{2 - \alpha \sqrt{2} \sin(3u) \sin(2v)}, \text{ and} \\z(u, v) &= \frac{3 \cos^2 v}{2 - \alpha \sqrt{2} \sin(3u) \sin(2v)}.\end{aligned}$$

Tables and lists are discussed in more detail in Chapters 4 and 5.

Here, $\alpha = 0$ gives the Roman surface and $\alpha = 1$ gives the Boy surface.

To see the homotopy we first define x , y , and z .

```
x[α-][u-, v-] = (Sqrt[2]Cos[2u]Cos[v]^2 + Cos[u]Sin[2v])/
(2 - α Sqrt[2]Sin[3u]Sin[2v]);
y[α-][u-, v-] = (Sqrt[2]Sin[2u]Cos[v]^2 + Sin[u]Sin[2v])/
(2 - α Sqrt[2]Sin[3u]Sin[2v]);
z[α-][u-, v-] = 3Cos[v]^2/
(2 - α Sqrt[2]Sin[3u]Sin[2v]);
```

We then use `Table` together with `ParametricPlot3D` to parametrically plot x , y , and z , $0 \leq u \leq 2\pi$, $0 \leq v \leq 2\pi$ for nine equally spaced values of α between 0 and 1. Note that if the semicolon is omitted at the end of the command, the nine plots are displayed.

```
smalltable = Table[ParametricPlot3D[
  {x[α][u, v], y[α][u, v], z[α][u, v]},
  {u, 0, 2Pi}, {v, 0, 2Pi}, Boxed → False, Axes → None,
  PlotRange → {{-2, 5/2}, {-2, 2}, {0, 7/2}},
  {α, 0, 1, 1/8}];
```

We then use `Partition` to partition `smalltable` into three element subsets. The resulting 3×3 array of graphics is shown as a grid with `Show` together with `GraphicsGrid` in Figure 2.30.

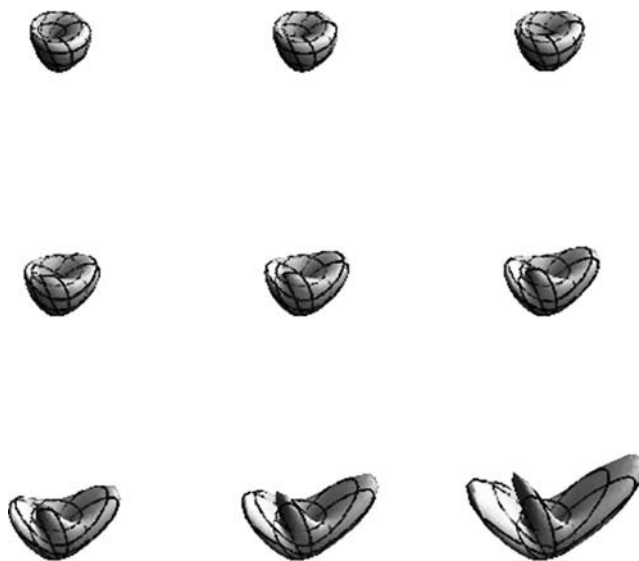
```
Show[GraphicsGrid[Partition[smalltable, 3]]]
```

Another way of seeing the transformation is to use `Manipulate`. `Manipulate` is very powerful.

To adjust the viewing angles of three-dimensional plots, select the graphic and drag to the desired viewing angle.

```
In[1]: ?Manipulate
```

```
Manipulate[expr, {u, u_min, u_max}] generates a version
of expr with controls added to allow interactive manipulation of the value of u.
Manipulate[expr, {u, u_min, u_max, du}] allows the value of u to vary between u_min and u_max in steps du.
Manipulate[expr, {{u, u_init}, u_min, u_max, ...}] takes the initial value of u to be u_init.
Manipulate[expr, {{u, u_init, u_init}, ...}] labels the controls for u with u_init.
Manipulate[expr, {u, {u1, u2, ...}}] allows u to take on discrete values u1, u2, ...
Manipulate[expr, {u, ...}, {v, ...}, ...] provides controls to manipulate each of the u, v, ...
Manipulate[expr, cu -> {u, ...}, cv -> {v, ...}, ...]
links the controls to the specified controllers on an external device. >>
```

**FIGURE 2.30**

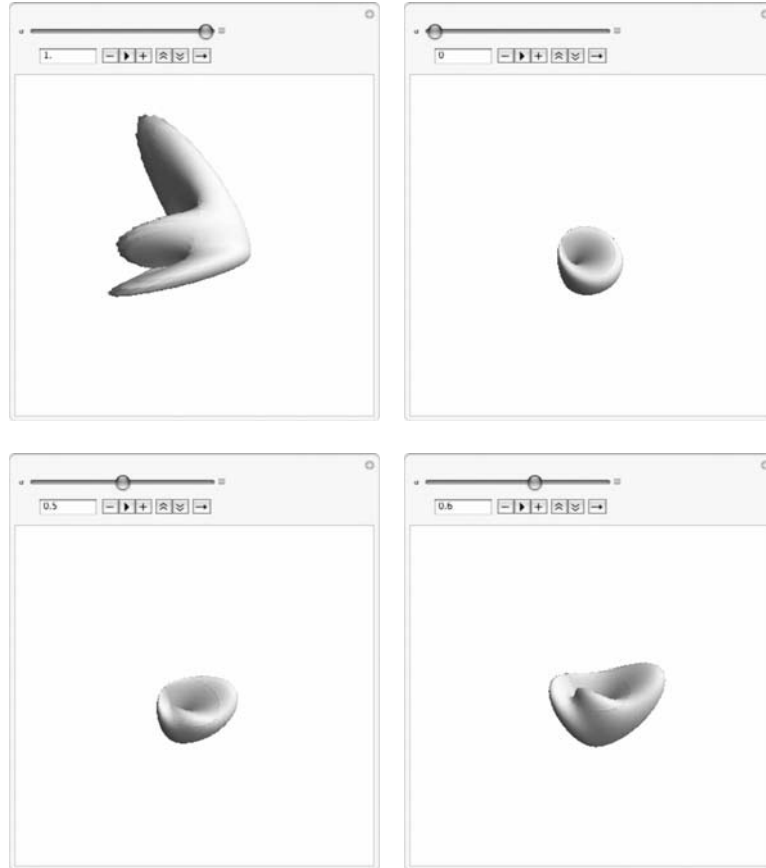
Seeing the Roman surface continuously transform to the Boy surface

In its most basic form, `Manipulate[f[x],{x,a,b}]` creates an interactive display of $f(x)$ for x values from a to b . Because the previous commands depended only on α , we combine the commands into a single `Manipulate` object that depends on α .

```
Manipulate[
Clear[x, y, z];
 $x[\alpha\_][u\_ , v\_ ] = (\text{Sqrt}[2]\text{Cos}[2u]\text{Cos}[v]^2 + \text{Cos}[u]\text{Sin}[2v]) /$ 
 $(2 - \alpha\text{Sqrt}[2]\text{Sin}[3u]\text{Sin}[2v]);$ 
 $y[\alpha\_][u\_ , v\_ ] = (\text{Sqrt}[2]\text{Sin}[2u]\text{Cos}[v]^2 + \text{Sin}[u]\text{Sin}[2v]) /$ 
 $(2 - \alpha\text{Sqrt}[2]\text{Sin}[3u]\text{Sin}[2v]);$ 
 $z[\alpha\_][u\_ , v\_ ] = 3\text{Cos}[v]^2 /$ 
 $(2 - \alpha\text{Sqrt}[2]\text{Sin}[3u]\text{Sin}[2v]);$ 
ParametricPlot3D[
 $\{x[\alpha][u, v], y[\alpha][u, v], z[\alpha][u, v]\},$ 
 $\{u, 0, 2\text{Pi}\}, \{v, 0, 2\text{Pi}\}, \text{PlotPoints} \rightarrow 50,$ 
 $\text{Boxed} \rightarrow \text{False}, \text{Axes} \rightarrow \text{None},$ 
 $\text{PlotRange} \rightarrow \{\{-2.5, 2.5\}, \{-2.5, 2.5\}, \{0, 3.5\}\},$ 
 $\{\alpha, 0, 1\}$ 
]
```

Several images from the result are shown in Figure 2.31.

Manipulation of graphics is discussed in more detail in Chapter 5. Here, we simply illustrate a few quick ways to manipulate a basic jpeg that illustrates a few of the features of Mathematica 6.

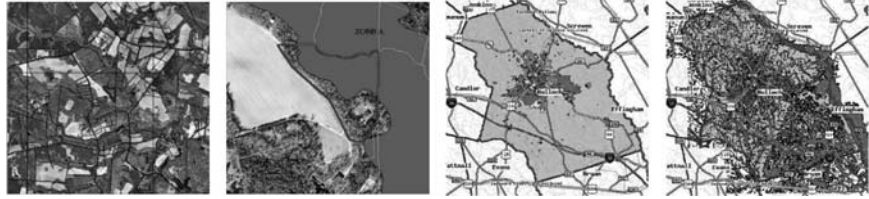
**FIGURE 2.31**

With **Manipulate** we can create an animation of the transformation of the Roman surface to the Boy surface or inspect the plot for various values of α

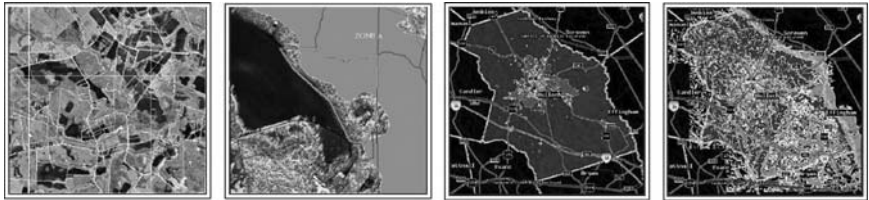
Example 2.3.22 We use **Import** to import a few graphics into Mathematica. The four graphs are displayed in a row using **Show** and **GraphicsRow** in Figure 2.32.

```
p1 = Import["house1.jpg"];
p2 = Import["house2.jpg"];
p3 = Import["county1.jpg"];
p4 = Import["county2.jpg"];
Show[GraphicsRow[{p1, p2, p3, p4}]]
```

The underlying structure of a jpeg is contained in the first element of the first part of the graphic. Part and manipulation of matrices is discussed in more detail in

**FIGURE 2.32**

Importing elementary graphics into Mathematica

**FIGURE 2.33**

Visualizing data with `ArrayPlot` may not produce results identical to the original without additional adjustments

Chapter 5. Regardless, you should be able to import your jpeg into Mathematica and adjust the following code to achieve your desired results.

With just the basic data, `ArrayPlot` does not reproduce the imported graphics as we see in Figure 2.33.

```
Show[GraphicsRow[
  Map[ArrayPlot[Reverse[#[[1, 1]]]&, {p1, p2, p3, p4}]]]
```

However, with some manipulation, you can reveal interesting detail. First, we use `ReliefPlot` to help us see the terrain of the image.

```
q1 = Flatten[p1[[1, 1]], 1];
q1b = Table[q1[[i, 1]], {i, 1, Length[q1]}];
q1c = Partition[q1b, Length[p1[[1, 1]]]];

r1 = ReliefPlot[q1c, ColorFunction -> "GreenBrownTerrain"]
```

A different view is obtained by choosing a different `ColorFunction`.

```
q2 = Flatten[p2[[1, 1]], 1];
q2b = Table[q2[[i, 2]], {i, 1, Length[q2]}];
q2c = Partition[q2b, Length[p2[[1, 1]]]];

r2 = ReliefPlot[q2c, ColorFunction -> "GrayTones"]
```

`ReliefPlot` and `ArrayPlot` return similar graphics. Here are two images of Southeast Georgia generated with `ArrayPlot`.

```

q3 = Flatten[p3[[1, 1]], 1];
q3b = Table[q3[[i, 1]], {i, 1, Length[q3]}];
q3c = Partition[q3b, Length[p3[[1, 1, 1]]]];

r3 = ArrayPlot[Reverse[q3c], ColorFunction -> "FallColors"]

q4 = Flatten[p4[[1, 1]], 1];
q4b = Table[q4[[i, 1]], {i, 1, Length[q4]}];
q4c = Partition[q4b, Length[p4[[1, 1, 1]]]];

r4 = ArrayPlot[Reverse[q4c], ColorFunction -> "StarryNightColors"]

```

All four images are displayed together using Show with GraphicsGrid in Figure 2.34.

```
Show[GraphicsGrid[{{r1, r2}, {r3, r4}}]]
```

2.3.5 Miscellaneous Comments

Be sure to take advantage of **MathWorld** for a huge number of resources related to graphics and Mathematica.

Clearly, Mathematica's graphics capabilities are extensive and *volumes* could be written about them. You can see many commands that we have

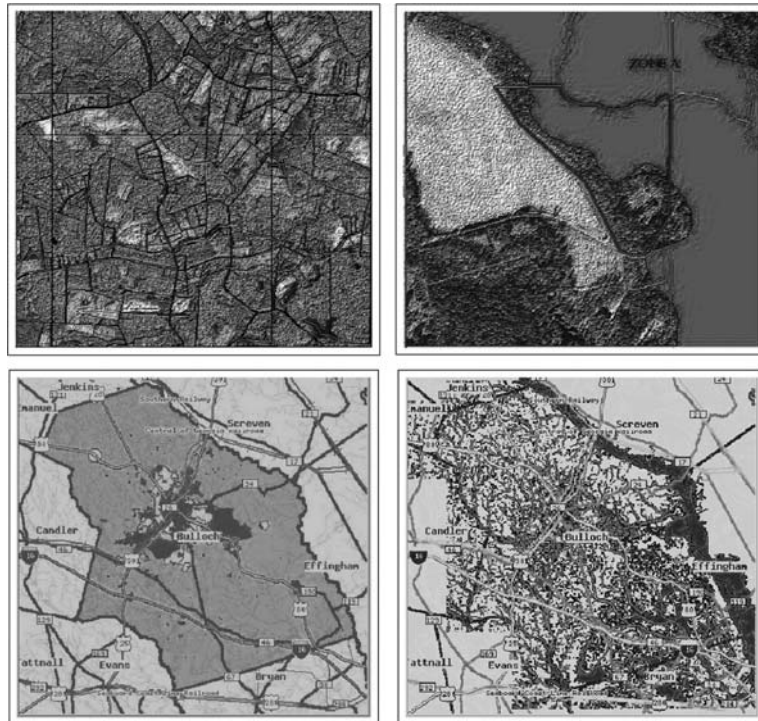


FIGURE 2.34

Using ReliefPlot and ArrayPlot to adjust elementary graphics

not discussed here by using ? to see those commands that contain the string Plot.

In[10]:= ?*Plot*

▼ System*

ArrayPlot	ListPlot3D	PlotLabel
ContourPlot	ListPointPlot3D	PlotMarkers
ContourPlot3D	ListPolarPlot	PlotPoints
DateListPlot	ListSurfacePlot3D	PlotRange
DensityPlot	LogLinearPlot	PlotRangeClipping
GraphPlot	LogLogPlot	PlotRangePadding
GraphPlot3D	LogPlot	PlotRegion
LayeredGraphPlot	MatrixPlot	PlotStyle
ListContourPlot	MaxPlotPoints	PolarPlot
ListContourPlot3D	ParametricPlot	RegionPlot
ListDensityPlot	ParametricPlot3D	RegionPlot3D
ListLinePlot	Plot	ReliefPlot
ListLogLinearPlot	Plot3D	RevolutionPlot3D
ListLogLogPlot	Plot3Matrix	SphericalPlot3D
ListLogPlot	PlotDivision	TreePlot
ListPlot	PlotJoined	

You can obtain detailed information regarding any of these commands from the **Documentation Center** by clicking on the command's name.

For now, we briefly mention a few of the ones not discussed previously. To plot lists of numbers or lists of ordered pairs, use ListPlot, which is discussed in Chapter 4. For matrices and other arrays, use MatrixPlot or ArrayPlot.

Example 2.3.23 (Cellular Automaton). Very loosely speaking, a **cellular automaton** is a discrete function that assigns values to subsequent rows based on the values of the cells in the previous row(s). For a concise discussion of cellular automaton, refer to Weisstein.¹ CellularAutomatan is a powerful command that allows you to investigate (quite complicated) cellular automaton. In its simplest form, CellularAutomaton[rule, initialvalues, n] returns the first n generations of the cellular automaton following the specified rule and having the indicated initial values.

The simplest cellular automaton are called **elementary cellular automaton**.² Based on basic counting principals, there are 256 elementary cellular automatons. They are cataloged by number. With

¹ Weisstein, Eric W., "Cellular Automaton." From *MathWorld—A Wolfram Web Resource*, <http://mathworld.wolfram.com/CellularAutomaton.html>.

² Weisstein, Eric W., "Elementary Cellular Automaton." From *MathWorld—A Wolfram Web Resource*, <http://mathworld.wolfram.com/ElementaryCellularAutomaton.html>.

CellularAutomaton[146, {{1}, 0}, 5]

```
{ {0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0}, {0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0},
  {0, 1, 0, 0, 0, 0, 0, 0, 1, 0}, {1, 0, 1, 0, 0, 0, 0, 1, 0, 1} }
```

we calculate the first five generations of the elementary cellular automaton with a 1 at position 0 on generation 0 using Rule 146. To calculate the first 100 generations, we use `CellularAutomaton[146, 1, 0, 100]`. The resulting array is rather large, so we use `ArrayPlot` to visualize it in Figure 2.35(a). Using our color scheme, the cells with value 1 are shaded red and those with 0 are light green.

**a1 = ArrayPlot[CellularAutomaton[146, {{1}, 0}, 100],
ColorFunction → “NeonColors”, AspectRatio → 1]**

In this case, the grid is initially spaced so that positions 1, 11, 21, 31, and 41 have the value 1. The first three generations using Rule 146 are calculated.

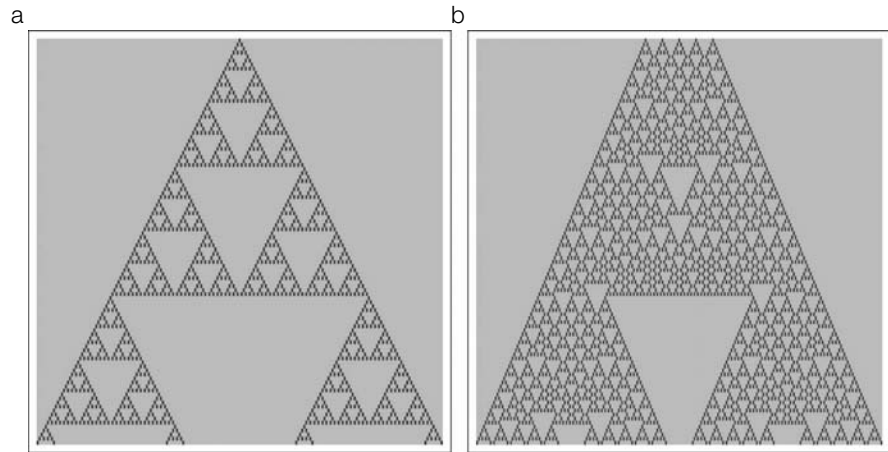
```
CellularAutomaton[146,
  {SparseArray[{1 → 1, 11 → 1, 21 → 1, 31 → 1, 41 → 1}], 0}, 3]
{ {0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
  {0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
    0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0},
  {0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
  1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0},
  {1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1,
    0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1} }
```

How the situation evolves over 100 generations is more easily seen using `ArrayPlot`. See Figure 2.35(b).

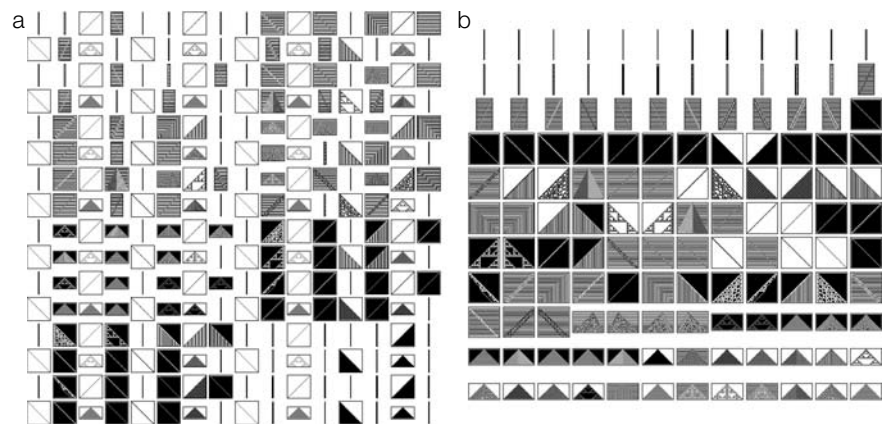
**a2 = ArrayPlot[CellularAutomaton[146,
{SparseArray[{1 → 1, 11 → 1, 21 → 1, 31 → 1, 41 → 1}], 0}, 100],
ColorFunction → “NeonColors”, AspectRatio → 1]
Show[GraphicsRow[{a1, a2}]]**

Of the 256 elementary cellular automaton, many are equivalent. To see that some of them are equivalent, we create a plot of the 256 elementary cellular automaton for 50 generations as done with Rule 146. All 256 plots are shown on the left in Figure 2.36(a). With `Union`, we remove and sort the ones that are identically equal. Those are shown on the right in Figure 2.36(b).

```
t1 = Table[ArrayPlot[CellularAutomaton[i, {{1}, 0}, 50]],
  {i, 0, 255}];
t2 = Partition[t1, 16];
p1 = Show[GraphicsGrid[t2]];
t3 = Union[t1];
t4 = Partition[t3, 12];
```

**FIGURE 2.35**

The evolution of two cellular automaton evolving according to Rule 146

**FIGURE 2.36**

(a) The first 50 generations for the 256 elementary cellular automaton. (b) Removal of the identical ones

```
p2 = Show[GraphicsGrid[t4];  
Show[GraphicsRow[{p1, p2}]]
```

To see the plots together with the rule number, use **Table**. Each order pair returned consists of the rule number and the 50 generation plot. To display the ordered pairs in an organized fashion, we use **Grid**. Of course, the result is quite large, so just a portion of the actual grid is displayed in Figure 2.37.

```
t5 = Table[{i, ArrayPlot[CellularAutomaton[i, {{1}, 0}, 50]],  
          {i, 0, 255}];
```

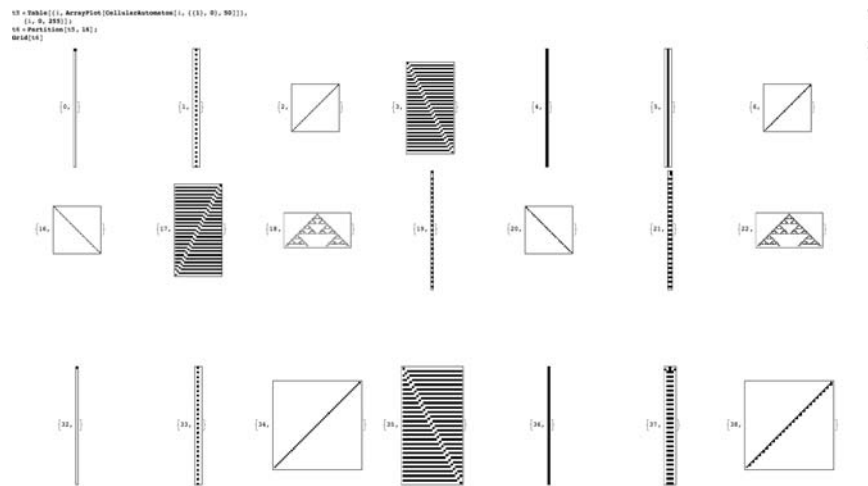


FIGURE 2.37

Seeing the automaton together with its rule number

t6 = Partition[t5, 16];
Grid[t6]

Note that `MatrixPlot` and `ArrayPlot` are discussed in more detail in Chapter 5.

For graphs of the form points or nodes connected by edges (graph theory), you can use `GraphPlot` to help investigate some problems. For trees, use `TreePlot`.

GraphPlot

New in 6

`GraphPlot[{v1 -> v2, v3 -> v4, ...}]`
generates a plot of the graph in which vertex v_1 is connected to vertex v_2 .

`GraphPlot[{v1 -> v2, M1}, ...]`
associates labels M_1 with edges in the graph.

`GraphPlot[m]`
generates a plot of the graph represented by the adjacency matrix m .

More Information

- `GraphPlot` attempts to place vertices to give a well laid-out version of the graph.
- The vertex names v_i and the labels M_i can be any expressions.
- With the default setting `DirectedEdges -> False`, edges are drawn as ordinary lines. With `DirectedEdges -> True`, edges are drawn as arrows.
- `GraphPlot` has the same options as `Graphics`, with the following additions and changes:

Option	Default	Description
<code>DirectedEdges</code>	<code>False</code>	whether to show edges as directed arrows
<code>EdgeLabeling</code>	<code>True</code>	whether to include labels given for edges
<code>EdgeRenderingFunction</code>	<code>Automatic</code>	function to give explicit graphics for edges
<code>Frame</code>	<code>False</code>	whether to draw a frame around the plot
<code>FrameTicks</code>	<code>None</code>	what ticks to include on the frame
<code>Method</code>	<code>Automatic</code>	the method used to lay out the graph
<code>MethodOptions</code>	<code>Automatic</code>	how to draw multiple edges between vertices
<code>PlottingMethod</code>	<code>Automatic</code>	method to use for packing components
<code>PlotRangePadding</code>	<code>Automatic</code>	how much padding to put around the plot
<code>PlotStyle</code>	<code>Automatic</code>	overall graphics directives for vertices and edges

TreePlot

New in 6

`TreePlot[{v1 -> v2, v3 -> v4, ...}]`
generates a tree plot of the graph in which vertex v_1 is connected to vertex v_2 .

`TreePlot[{v1 -> v2, M1}, ...]`
associates labels M_1 with edges in the graph.

`TreePlot[g, pos]`
places roots of trees in the plot at position pos .

`TreePlot[g, pos, u]`
uses vertex u as the root node in the tree plot.

More Information

- `TreePlot[g, ...]` lays out the vertices of g in a tree of successive layers, or a collection of trees.
- If the graph g is not a tree, `TreePlot` lays out its vertices on the basis of a spanning tree of each component of the graph.
- `TreePlot[g]` attempts to choose the root so as to make trees have as few layers as possible.
- By default, `TreePlot` places each tree root at the top. `TreePlot[g, pos]` places the roots at position pos . Possible positions are: `Top`, `Bottom`, `Left`, `Right`, `Center`.
- The vertex names v_i and the labels M_i can be any expressions.
- With the default setting `DirectedEdges -> False`, edges are drawn as ordinary lines. With `DirectedEdges -> True`, edges are drawn as directed arrows.
- `TreePlot` has the same options as `Graphics`, with the following additions and changes:

Option	Default	Description
<code>DirectedEdges</code>	<code>False</code>	whether to show edges as directed arrows

Example 2.3.24 Graceful graphs do not have multiple edges or loops.

We generate O with `GraphPlot` and display the result in Figure 2.38(a).

```
gp1 = GraphPlot[{{0->12, "12"}, {12->1, "11"}, {1->0, "1"}, {0->9, "9"},
  {1->9, "8"}, {1->6, "5"}, {12->6, "6"}, {2->12, "10"},
  {0->2, "2"}, {6->9, 9->10, 10->6},
  VertexLabeling -> True, AspectRatio -> 1]
```

C_4 is shown in Figure 2.38(b).

```
gp2 = GraphPlot[{0 -> 4, 4 -> 2, 2 -> 3, 3 -> 0}, DirectedEdges -> True,
  VertexLabeling -> True, AspectRatio -> 1]
```

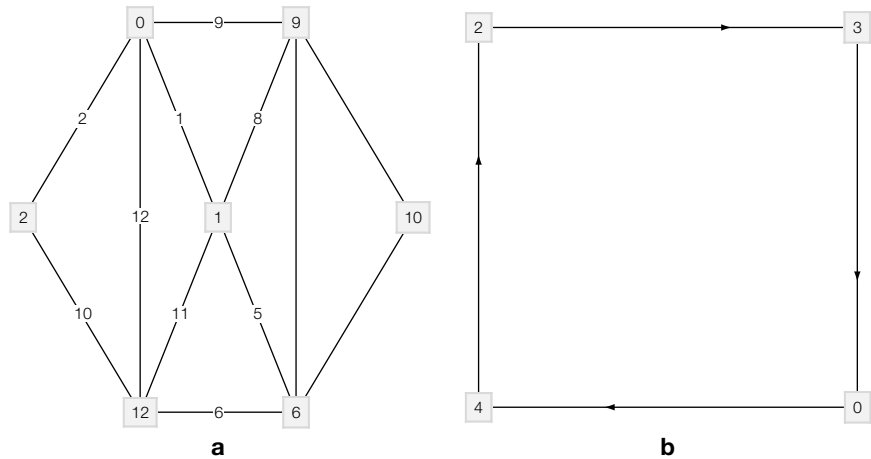


FIGURE 2.38

(a) O . (b) C_4

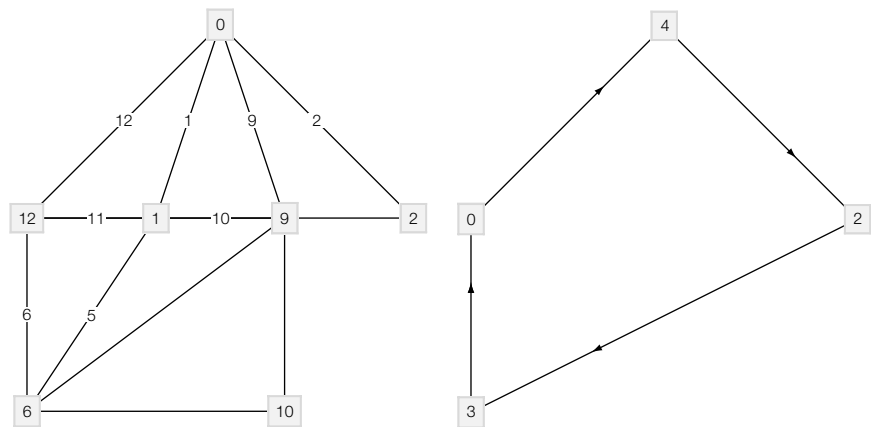


FIGURE 2.39

Using `TreePlot` instead of `GraphPlot`

Replacing GraphPlot with TreePlot gives us Figure 2.39.

```
tp1 = TreePlot[{{0 -> 12, "12"}, {12 -> 1, "11"}, {1 -> 0, "1"}, {0 -> 9, "9"},
  {1 -> 9, "8"}, {1 -> 6, "5"}, {12 -> 6, "6"}, {2 -> 12, "10"},
  {0 -> 2, "2"}, {6 -> 9, 9 -> 10, 10 -> 6},
  VertexLabeling -> True, AspectRatio -> 1]
tp2 = TreePlot[0 -> 4, 4 -> 2, 2 -> 3, 3 -> 0], DirectedEdges -> True,
  VertexLabeling -> True, AspectRatio -> 1]
```

2.4 SOLVING EQUATIONS

2.4.1 Exact Solutions of Equations

Mathematica can find exact solutions to many equations and systems of equations, including exact solutions to polynomial equations of degree four or less. Because a single equals sign “=” is used to name objects and assign values in Mathematica, equations in Mathematica are of the form

left-hand side==right-hand side.

The “double-equals” sign “==” between the left-hand side and right-hand side specifies that the object is an equation. For example, to represent the equation $3x + 7 = 4$ in Mathematica, type $3x+7==4$. The command `Solve[lhs==rhs,x]` solves the equation $\text{lhs} = \text{rhs}$ for x . If the only unknown in the equation $\text{lhs} = \text{rhs}$ is x and Mathematica does not need to use inverse functions to solve for x , the command `Solve[lhs==rhs]` solves the equation $\text{lhs} = \text{rhs}$ for x . Hence, to solve the equation $3x + 7 = 4$, both the commands `Solve[3x+7==4]` and `Solve[3x+7==4, x]` return the same result.



Example 2.4.1 Solve the equations $3x + 7 = 4$, $(x^2 - 1)/(x - 1) = 0$ and $x^3 + x^2 + x + 1 = 0$.

Solution In each case, we use **Solve** to solve the indicated equation. Be sure to include the double equals sign “==” between the left- and right-hand sides of each equation. Thus, the result of entering

Solve[3x + 7 == 4]
 $\{\{x \rightarrow -1\}\}$

means that the solution of $3x + 7 = 4$ is $x = -1$, and the result of entering

Solve $\left[\frac{x^2 - 1}{x - 1} == 0\right]$
 $\{\{x \rightarrow -1\}\}$

means that the solution of $\frac{x^2 - 1}{x - 1} = 0$ is $x = -1$. On the other hand, the equation $x^3 + x^2 + x + 1 = 0$ has two imaginary roots. We see that entering

Solve $[x^3 + x^2 + x + 1 == 0]$
 $\{\{x \rightarrow -1\}, \{x \rightarrow -i\}, \{x \rightarrow i\}\}$

yields all three solutions. Thus, the solutions of $x^3 + x^2 + x + 1 = 0$ are $x = -1$ and $x = \pm i$. Remember that the Mathematica symbol **I** represents the complex number $i = \sqrt{-1}$. In general, Mathematica can find the exact solutions of any polynomial equation of degree four or less.

Lists and tables are discussed in more detail in Chapters 4 and 5.

Observe that the results of a **Solve** command are a **list**.

Mathematica can also solve equations involving more than one variable for one variable (literal equations) in terms of other unknowns.

Example 2.4.2 (a) Solve the equation $v = \pi r^2/h$ for h . (b) Solve the equation $a^2 + b^2 = c^2$ for a .

Solution These equations involve more than one unknown, so we must specify the variable for which we are solving in the **Solve** commands. Thus, entering

Solve[v == Pi r^2/h, h]
 $\{\{h \rightarrow \frac{\pi r^2}{v}\}\}$

solves the equation $v = \pi r^2/h$ for h . (Be sure to include a space or * between π and r .) Similarly, entering

Solve[a^2 + b^2 == c^2, a]
 $\{\{a \rightarrow -\sqrt{-b^2 + c^2}\}, \{a \rightarrow \sqrt{-b^2 + c^2}\}\}$

solves the equation $a^2 + b^2 = c^2$ for a .

If Mathematica needs to use inverse functions to solve an equation, you must be sure to specify the variable(s) for which you want Mathematica to solve.

Example 2.4.3 Find a solution of $\sin^2 x - 2 \sin x - 3 = 0$.

Solution When the command `Solve[Sin[x]^2-2Sin[x]-3==0]` is entered, Mathematica solves the equation for `Sin[x]`. However, when the command

`Solve[Sin[x]^2-2Sin[x]-3==0, x]`

is entered, Mathematica attempts to solve the equation for x . In this case, Mathematica succeeds in finding one solution.

`Solve[Sin[x]^2-2Sin[x]-3==0]`

`{{Sin[x] → -1}, {Sin[x] → 3}}`

In fact, this equation has infinitely many solutions of the form $x = \frac{1}{2}(4k-1)\pi$, $k = 0, \pm 1, \pm 2, \dots$; $\sin x = 3$ has no solutions.

```
Solve[Sin[x]^2-2Sin[x]-3==0, x]
Solve::ifun :
Inverse functions are being used by Solve, so some solutions may
not be found; use Reduce for complete solution information. >>
{{x → -π/2}, {x → ArcSin[3]}}
```

The example indicates that it is especially important to be careful when dealing with equations involving trigonometric functions.

Example 2.4.4 Let $f(\theta) = \sin 2\theta + 2 \cos \theta$, $0 \leq \theta \leq 2\pi$. (a) Solve $f'(\theta) = 0$. (b) Graph $f(\theta)$ and $f'(\theta)$.

Solution After defining $f(\theta)$, we use `D` to compute $f'(\theta)$ and then use `Solve` to solve $f'(\theta) = 0$.

`D[f[x],x]` computes $f'(x)$; `D[f[x],{x,n}]` computes $f^{(n)}(x)$.
Topics from calculus are discussed in more detail in Chapter 3.

`f[θ_] = Sin[2θ] + 2Cos[θ];`

`df = f'[θ]`

`2Cos[2θ]-2Sin[θ]`

```
Solve[df == 0, θ]
Solve::ifun :
Inverse functions are being used by Solve, so some solutions may not be
found; use Reduce for complete solution information. >>
{{θ → -π/2}, {θ → π/6}, {θ → 5π/6}}
```

Notice that $-\pi/2$ is not between 0 and 2π . Moreover, $\pi/6$ and $5\pi/6$ are *not* the only solutions of $f'(\theta) = 0$ between 0 and 2π . Proceeding by hand, we use the identity $\cos 2\theta = 1 - 2 \sin^2 \theta$ and factor

expression /. x->y
replaces all
occurrences of x in
expression by y.

$$\begin{aligned}2 \cos 2\theta - 2 \sin \theta &= 0 \\1 - 2 \sin^2 \theta - \sin \theta &= 0 \\2 \sin^2 \theta + \sin \theta - 1 &= 0 \\(2 \sin \theta - 1)(\sin \theta + 1) &= 0\end{aligned}$$

so $\sin \theta = 1/2$ or $\sin \theta = -1$. Because we are assuming that $0 \leq \theta \leq 2\pi$, we obtain the solutions $\theta = \pi/6$, $5\pi/6$, or $3\pi/2$. We perform the same steps with Mathematica.

```
s1 = TrigExpand[df]
2Cos[θ]2 - 2Sin[θ] - 2Sin[θ]2

s2 = s1/.Cos[θ]^2 -> 1 - Sin[θ]^2
-2Sin[θ] - 2Sin[θ]2 + 2 (1 - Sin[θ]2)

Factor[s2]
-2(1 + Sin[θ])(-1 + 2Sin[θ])
```

Finally, we graph $f(\theta)$ and $f'(\theta)$ with Plot in Figure 2.40. Note that the plot is drawn to scale because we include the option `AspectRatio->Automatic`.

```
p1 = Plot[{f[θ], df}, {θ, 0, 2π}, AspectRatio -> Automatic]
```

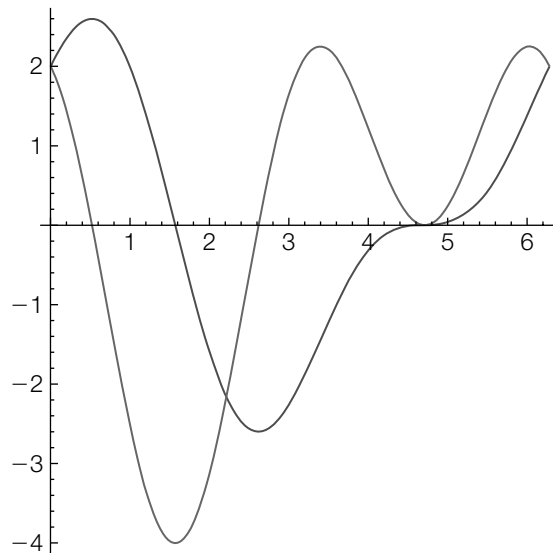


FIGURE 2.40

Graphs of $f(\theta)$ and $f'(\theta)$

We can also use `Solve` to find the solutions, if any, of various types of systems of equations. Entering

`Solve[{lhs1==rhs1, lhs2==rhs2}, {x, y}]`

solves a system of two equations for x and y , whereas entering

`Solve[{lhs1==rhs1, lhs2==rhs2}]`

attempts to solve the system of equations for all unknowns. In general, `Solve` can find the solutions to a system of linear equations. In fact, if the systems to be solved are inconsistent or dependent, Mathematica's output indicates so.

Example 2.4.5 Solve each system:

$$\begin{array}{ll}
 \text{(a)} \quad \begin{cases} 3x - y = 4 \\ x + y = 2 \end{cases} ; & \text{(b)} \quad \begin{cases} 2x - 3y + 4z = 2 \\ 3x - 2y + z = 0 \\ x + y - z = 1 \end{cases} ; & \text{(c)} \quad \begin{cases} 2x - 2y - 2z = -2 \\ -x + y + 3z = 0 \\ -3x + 3y - 2z = 1 \end{cases} ; \text{ and} \\
 \text{(d)} \quad \begin{cases} -2x + 2y - 2z = -2 \\ 3x - 2y + 2z = 2 \\ x + 3y - 3z = -3 \end{cases} .
 \end{array}$$

Solution In each case, we use `Solve` to solve the given system. For (a), the result of entering

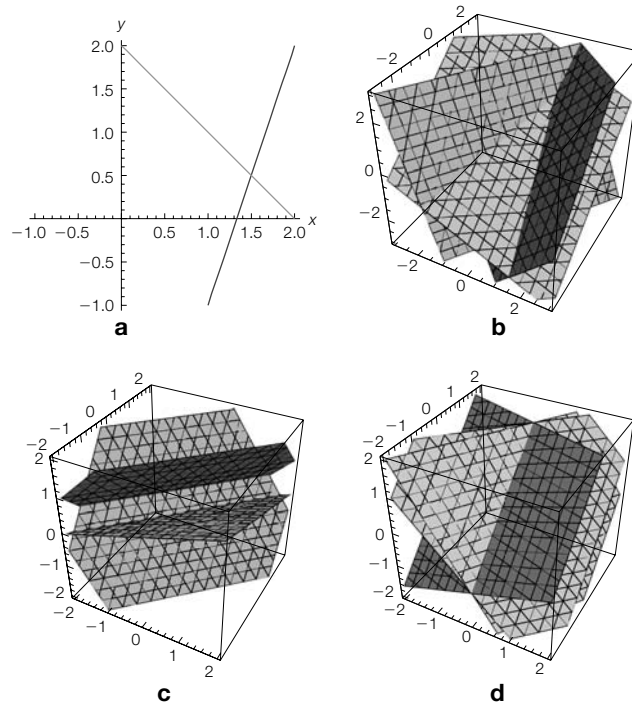
`Solve[{3x - y == 4, x + y == 2}, {x, y}]`
 $\left\{ \left\{ x \rightarrow \frac{3}{2}, y \rightarrow \frac{1}{2} \right\} \right\}$

means that the solution of $\begin{cases} 3x - y = 4 \\ x + y = 2 \end{cases}$ is $(x, y) = (3/2, 1/2)$, which is the point of intersection of the lines with equations $3x - y = 4$ and $x + y = 2$. See Figure 2.41(a).

**`cp1 = ContourPlot[{3x - y == 4, x + y == 2},`
**`{x, -1, 2}, {y, -1, 2}, Frame → False,`
**`Axes → Automatic, AxesOrigin → {0, 0},`
`AxesLabel → {x, y}] .1 in`******

(b) We can verify that the results returned by Mathematica are correct. First, we name the system of equations `sys` and then use `Solve` to solve the system of equations naming the result `sols`.

`sys = {2x - 3y + 4z == 2, 3x - 2y + z == 0,`
`x + y - z == 1};`
`sols = Solve[sys, {x, y, z}]`
 $\left\{ \left\{ x \rightarrow \frac{7}{10}, y \rightarrow \frac{9}{5}, z \rightarrow \frac{3}{2} \right\} \right\}$

**FIGURE 2.41**

(a) Two intersecting lines. (b) Three planes that intersect in a single point. (c) These three planes have no point in common. (d) The intersection of these three planes is a line

We verify the result by substituting the values obtained with `Solve` back into `sys` with `ReplaceAll (/.)`.

sys/.sols

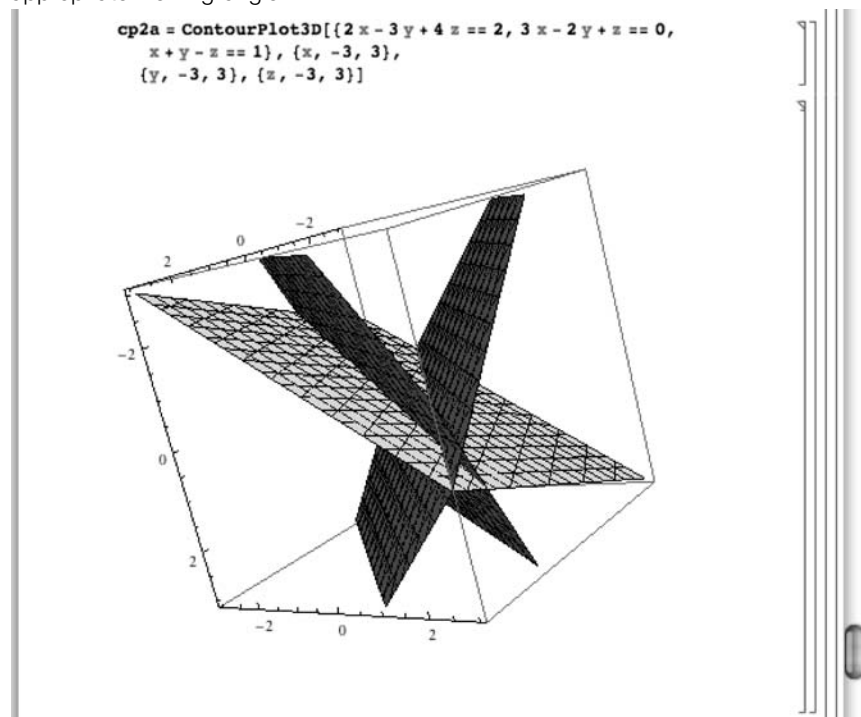
`{{True, True, True}}`

means that the solution of
$$\begin{cases} 2x - 3y + 4z = 2 \\ 3x - 2y + z = 0 \\ x + y - z = 1 \end{cases}$$
 is $(x, y, z) = (7/10, 9/5, 3/2)$, which

is the point of intersection of the planes with equations $2x - 3y + 4z = 2$, $3x - 2y + z = 0$, $x + y - z = 1$. See Figure 2.41(b).

```
cp2a = ContourPlot3D[{2x - 3y + 4z == 2, 3x - 2y + z == 0,  
x + y - z == 1}, {x, -3, 3},  
{y, -3, 3}, {z, -3, 3}]
```

To better see the intersection point, click within the graphic and then drag to an appropriate viewing angle.



(c) When we use `Solve` to solve this system, Mathematica returns `{}`, which indicates that the system has no solution; the system is inconsistent.

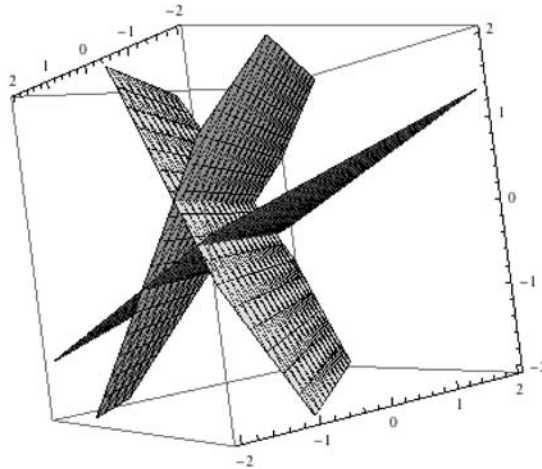
```
Solve[[2x - 2y - 2z == -2, -x + y + 3z == 0,
-3x + 3y - 2z == 1]]
{}
```

To see that the planes with equations $2x - 2y - 2z = -2$, $-x + y + 3z = 0$ and $-3x + 3y - 2z = 1$ have no points in common, graph them within Figure 2.41(c).

```
cp3a = ContourPlot3D[[2x - 2y - 2z == -2,
-x + y + 3z == 0, -3x + 3y - 2z == 1], {x, -2, 2},
{y, -2, 2}, {z, -2, 2}]
```

To better see that the planes do not intersect, we click and drag the graphic to an appropriate viewing angle.

```
cp3a = ContourPlot3D[{2 x - 2 y - 2 z == -2,
  -x + y + 3 z == 0, -3 x + 3 y - 2 z == 1}, {x, -2, 2},
  {y, -2, 2}, {z, -2, 2}]
```



(d) On the other hand, when we use **Solve** to solve this system, Mathematica's result indicates that the system has infinitely many solutions. That is, all ordered triples of the form $\{(0, z - 1, z) | z \text{ real}\}$ are solutions of the system.

```
Solve[{-2 x + 2 y - 2 z == -2, 3 x - 2 y + 2 z == 2,
  x + 3 y - 3 z == -3}]
Solve::svars :
Equations may not give solutions for all "solve" variables. >>
{{x -> 0, y -> -1 + z}}
```

We see that the intersection of the three planes is a line with **ContourPlot3D**. See Figure 2.41(d).

```
cp3a = ContourPlot3D[{2x - 2y - 2z == -2,
  3x - 2y + 2z == 2, x + 3y - 3z == -3}, {x, -2, 2},
  {y, -2, 2}, {z, -2, 2}]
Show[GraphicsGrid[{{cp1, cp2a}, {cp3a, cp4a}}]]
```

We can often use **Solve** to find solutions of a nonlinear system of equations as well.

Example 2.4.6 Solve the systems

$$(a) \begin{cases} 4x^2 + y^2 = 4 \\ x^2 + 4y^2 = 4 \end{cases} \quad \text{and (b)} \begin{cases} \frac{1}{a^2}x^2 + \frac{1}{b^2}y^2 = 1 \\ y = mx \end{cases} \quad (a, b \text{ greater than zero}) \text{ for } x \text{ and } y.$$

Solution The graphs of the equations are both ellipses. We use `ContourPlot` to graph each equation, naming the results `cp1` and `cp2`, respectively, and then use `Show` to display both graphs together in Figure 2.42(a). The solutions of the system correspond to the intersection points of the two graphs. Alternatively, the solutions of the system correspond to the intersection points of the level curves of $f(x, y) = 4x^2 + y^2 - 4$ and $g(x, y) = x^2 + 4y^2 - 4$ corresponding to 0. See Figure 2.42(b).

```
cp1 = ContourPlot [4x^2 + y^2 - 4, {x, -3, 3}, {y, -3, 3}, Contours -> {0},
  ContourShading -> False, PlotPoints -> 60 ];
cp2 = ContourPlot [x^2 + 4y^2 - 4, {x, -3, 3}, {y, -3, 3}, Contours -> {0},
  ContourShading -> False, PlotPoints -> 60 ];
cp3 = Show[cp1, cp2, Frame -> False, Axes -> Automatic, AxesOrigin -> {0, 0}]
cp4 = Plot3D [{4x^2 + y^2 - 4, x^2 + 4y^2 - 4}, {x, -10, 10},
  {y, -10, 10}, BoxRatios -> {1, 1, 1}, MeshFunctions -> {#3&},
  ColorFunction -> (ColorData["Rainbow"])[#3]&],
  PlotStyle -> {Opacity[.4], Opacity[.8]} ]
Show[GraphicsRow[{cp3, cp4}]]
```

Finally, we use `Solve` to find the solutions of the system.

$$\text{Solve} [4x^2 + y^2 == 4, x^2 + 4y^2 == 4]$$

$$\left\{ \left\{ x \rightarrow -\frac{2}{\sqrt{5}}, y \rightarrow -\frac{2}{\sqrt{5}} \right\}, \left\{ x \rightarrow -\frac{2}{\sqrt{5}}, y \rightarrow \frac{2}{\sqrt{5}} \right\}, \right. \\ \left. \left\{ x \rightarrow \frac{2}{\sqrt{5}}, y \rightarrow -\frac{2}{\sqrt{5}} \right\}, \left\{ x \rightarrow \frac{2}{\sqrt{5}}, y \rightarrow \frac{2}{\sqrt{5}} \right\} \right\}$$

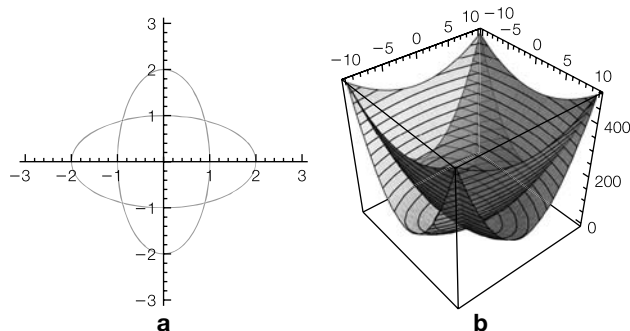


FIGURE 2.42

(a) Graphs of $4x^2 + y^2 = 4$ and $x^2 + 4y^2 = 4$. (b) Three-dimensional plots of $f(x, y)$ and $g(x, y)$ together with their level curves shown as contours

For (b), we also use **Solve** to find the solutions of the system. However, because the unknowns in the equations are a , b , m , x , and y , we must specify that we want to solve for x and y in the **Solve** command.

$$\text{Solve}\left[\left\{\frac{x^2}{a^2} + \frac{y^2}{b^2} == 1, y == mx\right\}, \{x, y\}\right]$$

$$\left\{\left\{y \rightarrow -\frac{abm}{\sqrt{b^2 + a^2 m^2}}, x \rightarrow -\frac{ab}{\sqrt{b^2 + a^2 m^2}}\right\}, \left\{y \rightarrow \frac{abm}{\sqrt{b^2 + a^2 m^2}}, x \rightarrow \frac{ab}{\sqrt{b^2 + a^2 m^2}}\right\}\right\}$$

Although Mathematica can find the exact solution to every polynomial equation of degree four or less, exact solutions to some equations may not be meaningful. In those cases, Mathematica can provide approximations of the exact solutions using either the **N[expression]** or the expression **// N** commands.

Example 2.4.7 Approximate the solutions to the equations (a) $x^4 - 2x^2 = 1 - x$; and (b) $1 - x^2 = x^3$.

Solution Each of these is a polynomial equation with degree less than five so **Solve** will find the exact solutions of each equation. However, the solutions are quite complicated, so we use **N** to obtain approximate solutions of each equation. For (a), entering

$$\text{N}[\text{Solve}[x^4 - 2x^2 == 1 - x]]$$

$$\{\{x \rightarrow 0.182777 - 0.633397i\}, \{x \rightarrow 0.182777 + 0.633397i\},$$

$$\{x \rightarrow -1.71064\}, \{x \rightarrow 1.34509\}\}$$

$$\{\{x \rightarrow 0.182777 - 0.633397i\},$$

$$\{x \rightarrow 0.182777 + 0.633397i\}, \{x \rightarrow -1.71064\}, \{x \rightarrow 1.34509\}\}$$

first finds the exact solutions of the equation $x^4 - 2x^2 = 1 - x$ and then computes approximations of those solutions. The resulting output is the list of approximate solutions. For (b), entering

$$\text{Solve}[1 - x^2 == x^3, x] // \text{N}$$

$$\{\{x \rightarrow 0.754878\}, \{x \rightarrow -0.877439 + 0.744862i\},$$

$$\{x \rightarrow -0.877439 - 0.744862i\}\}$$

$$\{\{x \rightarrow 0.754878\}, \{x \rightarrow -0.877439 + 0.744862i\},$$

$$\{x \rightarrow -0.877439 - 0.744862i\}\}$$

first finds the exact solutions of the equation $1 - x^2 = x^3$ and then computes approximations of those solutions. The resulting output is the list of approximate solutions.

2.4.2 Approximate Solutions of Equations

When solving an equation is either impractical or impossible, Mathematica provides several functions, including FindRoot, NRoots, and NSolve, to approximate solutions of equations. NRoots and NSolve numerically approximate the roots of any polynomial equation. The command NRoots[poly1{==}poly2, x] approximates the solutions of the polynomial equation poly1{==}poly2, where both poly1 and poly2 are polynomials in x . The syntax for NSolve is the same as the syntax of NRoots.

FindRoot attempts to approximate a root to an equation provided that a “reasonable” guess of the root is given. FindRoot works on functions other than polynomials. The command

FindRoot[lhs==rhs, {x, firstguess}]

searches for a numerical solution to the equation lhs==rhs, starting with x =firstguess. To locate more than one root, FindRoot must be used several times. One way of obtaining firstguess (for real-valued solutions) is to graph both lhs and rhs with Plot, find the point(s) of intersection, and estimate the x -coordinates of the point(s) of intersection. Generally, NRoots is easier to use than FindRoot when trying to approximate the roots of a polynomial.

Example 2.4.8 Approximate the solutions of $x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7 = 0$.

Solution Because $x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7 = 0$ is a polynomial equation, we may use NRoots to approximate the solutions of the equation. Thus, entering

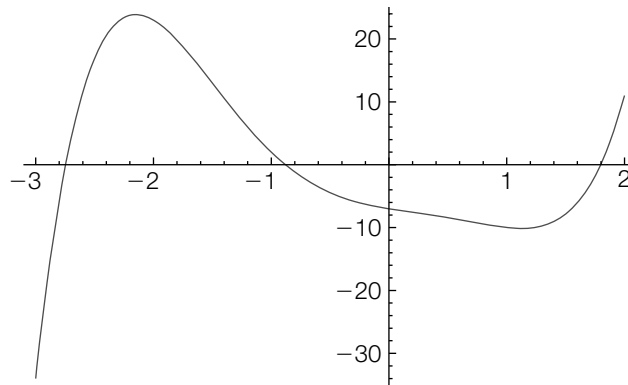
```
NRoots [x5 + x4 - 4x3 + 2x2 - 3x - 7==0, x]
x== -2.74463||x== -0.880858||x==0.41452 - 1.19996i||x==0.41452+
1.19996i||x==1.79645
x== -2.74463||x== -0.880858||x==0.41452 - 1.19996i||
x==0.41452 + 1.19996i||x==1.79645
```

approximates the solutions of $x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7 = 0$. The symbol || appearing in the result represents “or.”

We obtain equivalent results with NSolve.

```
NSolve [x5 + x4 - 4x3 + 2x2 - 3x - 7==0, x]
{{x→ -2.74463}, {x→ -0.880858}, {x→0.41452 - 1.19996i},
{x→0.41452 + 1.19996i}, {x→1.79645}}
{{x→ -2.74463}, {x→ -0.880858}, {x→0.41452 - 1.19996i},
{x→0.41452 + 1.19996i}, {x→1.79645}}
```

FindRoot may also be used to approximate each root of the equation. However, to use FindRoot, we must supply an initial approximation of the solution that we wish to approximate. The real solutions of $x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7 = 0$ correspond

**FIGURE 2.43**

Graph of $f(x) = x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7$

to the values of x where the graph of $f(x) = x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7$ intersects the x -axis. We use `Plot` to graph $f(x)$ in Figure 2.43.

Plot [$x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7$, { x , -3, 2}]

We see that the graph intersects the x -axis near $x \approx -2.5$, -1 , and 1.5 . We use these values as initial approximations of each solution. Thus, entering

FindRoot [$x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7 == 0$, { x , -2.5}]
{ $x \rightarrow -2.74463$ }

approximates the solution near -2.5 , entering

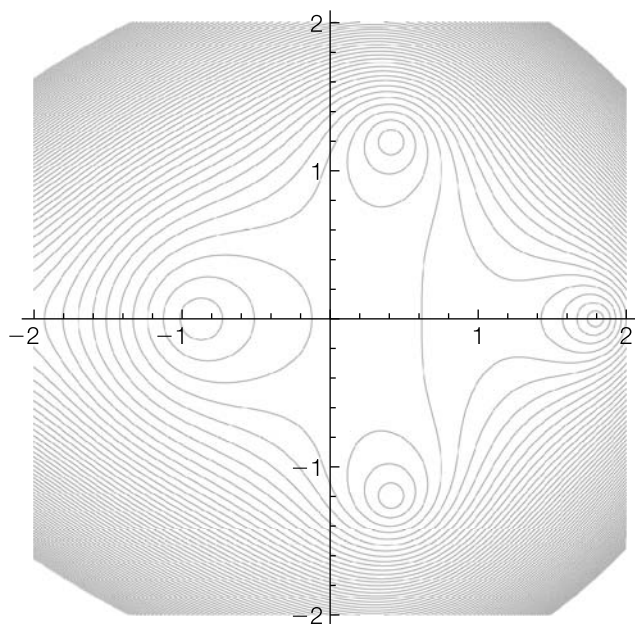
FindRoot [$x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7 == 0$, { x , -1}]
{ $x \rightarrow -0.880858$ }

approximates the solution near -1 , and entering

FindRoot [$x^5 + x^4 - 4x^3 + 2x^2 - 3x - 7 == 0$, { x , 2}]
{ $x \rightarrow 1.79645$ }

approximates the solution near 1.5 . Note that `FindRoot` may be used to approximate complex solutions as well. To obtain initial guesses, observe that the solutions of $f(z) = 0$, $z = x + iy$, x, y real, are the level curves of $w = |f(z)|$ that are points. In Figure 2.44, we use `ContourPlot` to graph various level curves of $w = |f(x + iy)|$, $-2 \leq x \leq 2$, $-2 \leq y \leq 2$. In the plot, observe that the two complex solutions occur at $x \pm iy \approx 0.5 \pm 1.2i$.

f[z_] = z⁵ + z⁴ - 4z³ + 2z² - 3z - 7;
ContourPlot[Abs[f[x + Iy]], {x, -2, 2}, {y, -2, 2},
ContourShading → False, Contours → 60,

**FIGURE 2.44**

Level curves of $w = |f(x + iy)|$, $-2 \leq x \leq 2$, $-2 \leq y \leq 2$

**PlotPoints $\rightarrow$ 200, Frame $\rightarrow$ False, Axes $\rightarrow$ Automatic,
AxesOrigin $\rightarrow$ {0, 0}}**

Thus, entering

```
FindRoot[x5 + x4 - 4x3 + 2x2 - 3x - 7 == 0, {x, 0.5 + I}]  
{x  $\rightarrow$  0.41452 + 1.19996i}
```

approximates the solution near $x + iy \approx 0.5 + 1.2i$. For polynomials with real coefficients, complex solutions occur in conjugate pairs so the other complex solution is approximately $0.41452 - 1.19996i$.

Example 2.4.9 Find the first three nonnegative solutions of $x = \tan x$.

Solution We attempt to solve $x = \tan x$ with Solve.

```
Solve[x == Tan[x], x]  
Solve::tdep : The equations appear to involve the  
variables to be solved for in an essentially non-algebraic way. >>  
Solve[x == Tan[x], x]
```



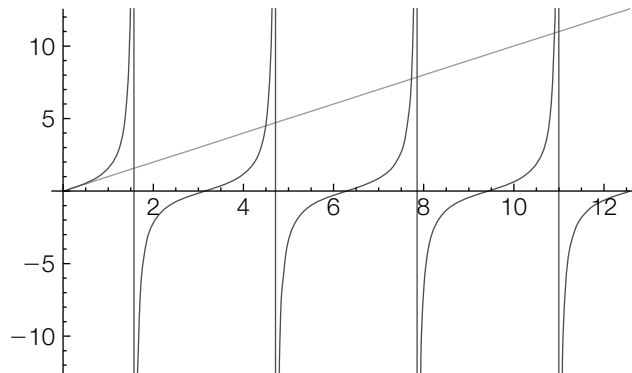


FIGURE 2.45

$y = x$ and $y = \tan x$

We next graph $y = x$ and $y = \tan x$ together in Figure 2.45.

```
Plot[Tooltip[{x, Tan[x]}], {x, 0, 4Pi},
     PlotRange -> {-4Pi, 4Pi}]
```

Remember that vertical lines are never the graphs of functions. In this case, they represent the vertical asymptotes at odd multiples of $\pi/2$.

In the graph, we see that $x = 0$ is a solution. This is confirmed with `FindRoot`.

```
FindRoot[x==Tan[x], {x, 0}]
{x -> 0.}
```

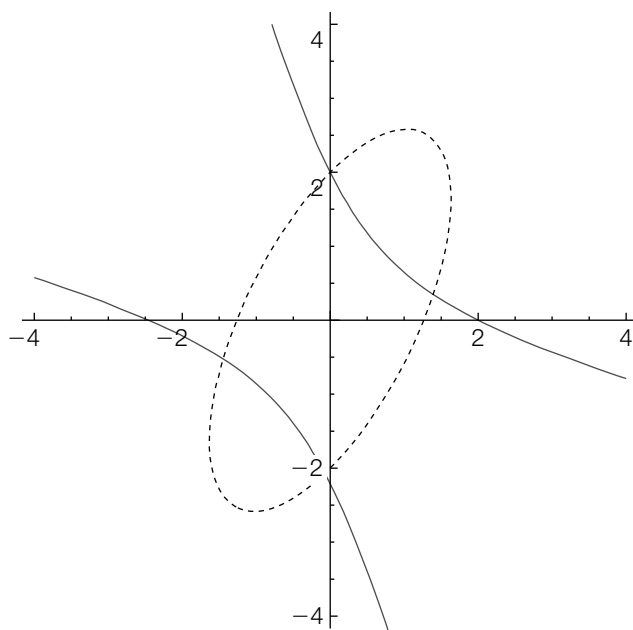
The second solution is near 4, whereas the third solution is near 7. Using `FindRoot` together with these initial approximations locates the second two solutions.

```
FindRoot[x==Tan[x], {x, 4}]
{x -> 4.49341}
FindRoot[x==Tan[x], {x, 7}]
{x -> 7.72525}
```

`FindRoot` can also be used to approximate solutions to systems of equations. (Although `NRroots` can solve a polynomial equation, `NRroots` cannot be used to solve a system of polynomial equations.) When approximations of solutions of systems of equations are desired, use either `Solve` and `N` together, when possible, or `FindRoot`.

Example 2.4.10 Approximate the solutions to the system of equations
$$\begin{cases} x^2 + 4xy + y^2 = 4 \\ 5x^2 - 4xy + 2y^2 = 8 \end{cases}.$$

Solution We begin by using `ContourPlot` to graph each equation in Figure 2.46. From the resulting graph, we see that $x^2 + 4xy + y^2 = 4$ is a hyperbola, $5x^2 - 4xy + 2y^2 = 8$ is an ellipse, and there are four solutions to the system of equations.

**FIGURE 2.46**

Graphs of $x^2 + 4xy + y^2 = 4$ and $5x^2 - 4xy + 2y^2 = 8$

```

cp1 = ContourPlot[x2 + 4xy + y2 - 4 == 0, {x, -4, 4}, {y, -4, 4},
  PlotPoints -> 60, ContourShading -> False];
cp2 = ContourPlot[5x2 - 4xy + 2y2 - 8 == 0, {x, -4, 4},
  {y, -4, 4}, PlotPoints -> 60, ContourStyle -> Dashing[{0.01}],
  ContourShading -> False];
Show[cp1, cp2, Frame -> False, Axes -> Automatic,
  AxesOrigin -> {0, 0}]

```

From the graph we see that possible solutions are $(0, 2)$ and $(0, -2)$. In fact, substituting $x = 0$ and $y = -2$ and $x = 0$ and $y = 2$ into each equation verifies that these points are both exact solutions of the equation. The remaining two solutions are approximated with `FindRoot`.

```

FindRoot[{x2 + 4xy + y2 == 4, 5x2 - 4xy + 2y2 == 8},
  {x, 1}, {y, 0.25}]
{x -> 1.39262, y -> 0.348155}
FindRoot[{x2 + 4xy + y2 == 4, 5x2 - 4xy + 2y2 == 8},
  {x, -1}, {y, -0.25}]
{x -> -1.39262, y -> -0.348155}

```

2.5 EXERCISES

1. Evaluate the following:

- (a) $432 + 701$
- (b) 251×8197
- (c) $\sqrt{116281}$
- (d) $\sqrt[3]{157464}$
- (e) $679/42$
- (f) $\sin(\pi/12)$
- (g) $\cos(11\pi/12)$
- (h) $\left| \frac{2+i}{5-3i} \right|$

- 2. Solve $x^3 - 8x^2 + 19x - 12 = \frac{1}{2}x^2 - x - \frac{1}{8}$. Confirm your result graphically.
- 3. Solve $-3x^2 + 12x - 5 = 2x^2 - 4x - 3$. Confirm your result graphically.
- 4. Find a 10-digit approximation of Euler's constant, denoted by EulerGamma.
- 5. Use TrigExpand to write $\sin 5x$ in terms of $\sin x$ and $\cos x$.
- 6. Use ExpToTrig to rewrite $e^x - e^{-x}$.
- 7. Use TrigToExp to rewrite $\cos it$.
- 8. Factor $15x^5 + 73x^4 - 621x^3 - 297x^2 + 2486x + 504$ to find the zeros of this polynomial. Compare these results with those obtained using Solve, FindRoot, or NRoots.
- 9. Use PowerExpand to simplify $\sqrt{a^4 b^6 c^{-8}}$. Compare this result with that obtained using Simplify and explain the difference.
- 10. Solve $\exp(-(x/4)^2) \cos(x/\pi) = \sin(x^{3/2}) + \frac{5}{4}$. Confirm your result graphically.
- 11. Graph the cross-cap, $4x^2(x^2 + y^2 + z^2 + z) + y^2(y^2 + z^2 - 1)$, by graphing it as functions of (a) y and z , (b) x and y , and (c) x and z .
- 12. Determine the partial fraction decomposition of the following.

- (a) $\frac{6x - 18}{x^2 - 2x - 8}$
- (b) $4 \frac{2x^2 + x + 28}{x^3 - 4x^2 + 16x - 64}$
- (c) $\frac{17x^2 + 2x}{x^4 + 5x^2 + 4}$

13. Let $f(x) = \begin{cases} -x, & x \leq -1 \\ \sin \pi x, & -1 < x \leq 1 \\ \frac{1}{2}x^2, & x > 1 \end{cases}$. Plot $f(x)$ and $f'(x)$ on the interval $[-5, 5]$.

14. Plot $\cos 2x$ and $2 \sin x$ on $-\pi \leq x \leq \pi$. Use FindRoot to determine where the two curves intersect on $[-\pi, \pi]$.
15. Consider Example 2.3.5 using $f(x) = \cos x$ with $n = 2, 4, \dots, 12$. Describe the graph of $f''(x)$ as n increases.
16. Graph the **tooth surface**, which is defined by the equation $x^4 + y^4 + z^4 - (x^2 + y^2 + z^2) = 0$.
17. Find a 20-digit approximation of Catalan's constant, denoted by Catalan.
18. Use ParametricPlot to graph (a) $x = 2 \cos t$, $y = 3 \sin t$, $0 \leq t \leq 2\pi$ and (b) $x = t \cos t/2$, $y = t \sin t/2$, $0 \leq t \leq 12\pi$.
19. Use PolarPlot to graph (a) $r = 2 \cos \theta$, $0 \leq \theta \leq \pi$ and (b) $r = \cos 2\theta$, $0 \leq \theta \leq 2\pi$.
20. Use PolarPlot to investigate the differences in the graph of $r = \cos n\theta$ and $r = \sin n\theta$ for n an odd or even integer. *Question:* What happens when n is a noninteger rational number?
21. Graph the level curves of the following: (a) $f(x, y) = x^2 - y^2$, (b) $f(x, y) = \sin(xy)$, and (c) $f(x, y) = x \cos y$.
22. Graph the function $f(x, y) = \sin(x^2 + y^2)$. Use the Interactive 3D control to rotate the graph in order to investigate the level curves of the function. Compare your findings to those obtained with ContourPlot.
23. A parametrization (X, Y, Z) of Boy's surface is given by

$$X = g_1/g \quad Y = g_2/g \quad Z = g_3/g,$$

where $g_1 = -\frac{3}{2} \operatorname{Im} \left(\frac{z(1 - z^4)}{z^6 + \sqrt{5}z^3 - 1} \right)$, $g_2 = -\frac{3}{2} \operatorname{Re} \left(\frac{z(1 + z^4)}{z^6 + \sqrt{5}z^3 - 1} \right)$,
 $g_3 = \operatorname{Im} \left(\frac{1 + z^6}{z^6 + \sqrt{5}z^3 - 1} \right) - \frac{1}{2}$, and $g = g_1^2 + g_2^2 + g_3^2$. The complex number $z = a + bi$ satisfies $|z| \leq 1$. Plot Boy's surface. (See Figure 2.47.)

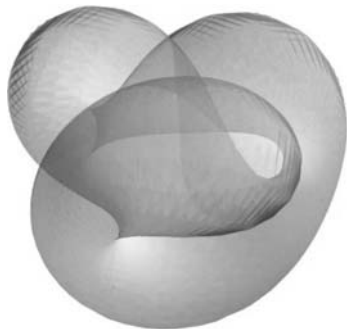


FIGURE 2.47

Boy's surface

Chapter 3 introduces Mathematica's built-in calculus functions. The examples used to illustrate the various functions are similar to examples typically seen in a traditional calculus sequence. If you have trouble typing commands correctly, use the buttons on the **BasicMathInput** palette to help you create templates in standard mathematical notation that you can evaluate.

3.1 LIMITS AND CONTINUITY

One of the first topics discussed in calculus is that of limits. Mathematica can be used to investigate limits graphically and numerically. In addition, the Mathematica command `Limit[f[x], x->a]` attempts to compute the limit of $y = f(x)$ as x approaches a , $\lim_{x \rightarrow a} f(x)$, where a can be a finite number, ∞ (Infinity), or $-\infty$ (-Infinity). The arrow " $\rightarrow$ " is obtained by typing a minus sign " $-$ " followed by a greater than sign " $>$ ".

Remark 3.1

`Clear[f]` clears all prior definitions of f , if any. Clearing function definitions before defining new ones helps eliminate any possible confusion and/or ambiguities.

To define a function of a single variable, $f(x) = \text{expression in } x$, enter `f[x_] = expression in x`. To generate a basic plot of $y = f(x)$ for $a \leq x \leq b$, enter `Plot[f[x], {x, a, b}]`.

3.1.1 Using Graphs and Tables to Predict Limits

Example 3.1.1 Use a graph and table of values to investigate $\lim_{x \rightarrow 0} \frac{\sin 3x}{x}$.

Solution We clear all prior definitions of f , define $f(x) = (\sin 3x)/x$, and then graph $y = f(x)$ on the interval $[-\pi, \pi]$ with `Plot`.



`RandomReal[{a,b}]` returns a “random” real number between a and b . Because we are generating “random” numbers, your results will differ from those obtained here.

`Map[f,{x1,x2,...,xn}]` returns the list $\{f(x_1), f(x_2), \dots, f(x_n)\}$.

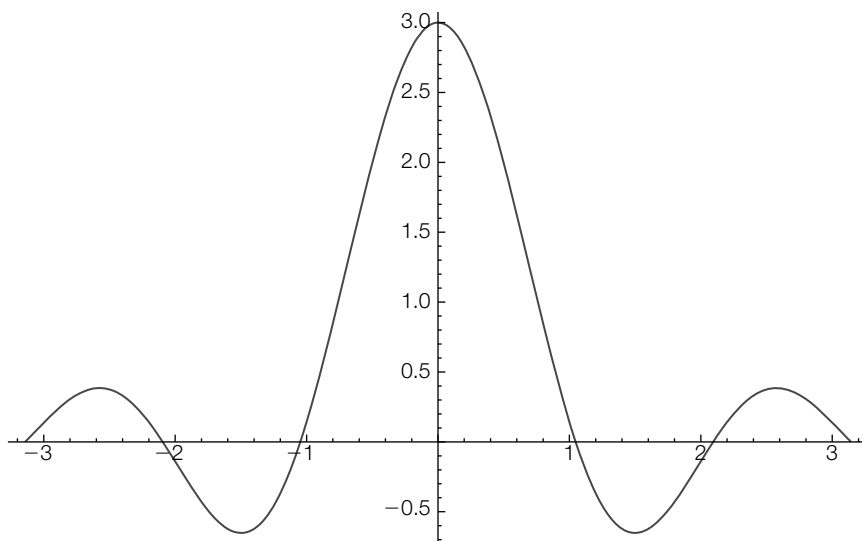


FIGURE 3.1

Graph of $f(x) = (\sin 3x)/x$ on the interval $[-\pi, \pi]$

```
Clear[f]
f[x_] = Sin[3x]/x;
Plot[f[x], {x, -pi, pi}]
```

From the graph shown in Figure 3.1, we might, correctly, conclude that $\lim_{x \rightarrow 0} \frac{\sin 3x}{x} = 3$. Further evidence that $\lim_{x \rightarrow 0} \frac{\sin 3x}{x} = 3$ can be obtained by computing the values of $f(x)$ for values of x “near” $x = 0$. In the following, we use `RandomReal` to define `xvals` to be a table of six “random” real numbers. The first number in `xvals` is between -1 and 1 , the second between $-1/10$ and $1/10$, and so on.

```
xvals = Table [RandomReal [{-10^-n, 10^-n}], {n, 0, 5}]
0.424046, -0.0850248, ...
```

We then use `Map` to compute the value of $f(x)$ for each x in `xvals`. We use `Table` to display the results in tabular form. Generally, `list[[i]]` returns the i th element of `list` while `Table[f[i], {i, start, finish, stepsize}]` computes each value of $f(i)$ from `start` to `finish` in increments of `stepsize`. `TableForm` attempts to display a table form in a standard format such as the row-and-column format that follows.

```
fvals = Map[f, xvals]
{2.25384, 2.96757, 2.99995, 3., 3., 3.}

pairs = Table[{xvals[[i]], fvals[[i]]}, {i, 1, 6}];
TableForm[pairs]
```

0.424046	2.25384
-0.0850248	2.96757
0.00334803	2.99995
0.0000981987	3.
0.0000376656	3.
-2.914605226592692 [^] -6	3.

From these values, we might again correctly deduce that $\lim_{x \rightarrow 0} \frac{\sin 3x}{x} = 3$. Of course, these results do not prove that $\lim_{x \rightarrow 0} \frac{\sin 3x}{x} = 3$, but they are helpful in convincing us that $\lim_{x \rightarrow 0} \frac{\sin 3x}{x} = 3$.

For piecewise-defined functions, you can either use Mathematica's *conditional command* (/;) to define the piecewise-defined function or use Piecewise.

Example 3.1.2 If $b(x) = \begin{cases} x^2 + x, & \text{if } x \leq 0 \\ 1 - x, & \text{if } 0 < x < 3 \\ 2x^2 - 15x + 25, & \text{if } 3 \leq x \leq 5 \\ 15 - 2x, & \text{if } x > 5 \end{cases}$, compute the following limits:
 (a) $\lim_{x \rightarrow 0} b(x)$, (b) $\lim_{x \rightarrow 3} b(x)$, (c) $\lim_{x \rightarrow 5} b(x)$.

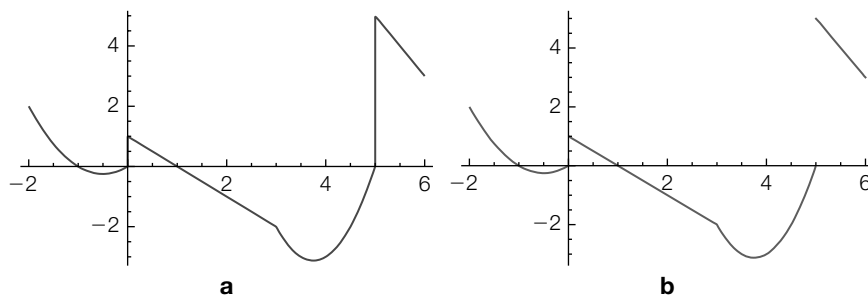
Solution We use Mathematica's conditional command, /;, to define b . We must use delayed evaluation (:=) because $b(x)$ cannot be computed unless Mathematica is given a particular value of x . The first line of the following defines $b(x)$ to be $x^2 + x$ for $x \leq 0$, the second line defines $b(x)$ to be $1 - x$ for $0 < x < 3$, and so on. In the Plot command, {x, -2, 0, 3, 5, 6} instructs Mathematica to graph the function on $[-2, 0]$, then $[0, 3]$, then $[3, 5]$, and finally $[5, 6]$. Notice that Mathematica accidentally connects (0, 0) to (0, 1) and then (5, 0) to (5, 5). (See Figure 3.2(a)). The delayed evaluation is also incompatible with Mathematica's Limit function.

The plots p1 and p2 are not displayed because a semicolon is included at the end of each Plot command.

```
Clear[h]
h[x_]:=x^2+x/x<=0
h[x_]:=1-x/0<x<3
h[x_]:=2x^2-15x+25/3<=x<=5
h[x_]:=15-2x/x>5
p1=Plot[h[x],{x,-2,0,3,5,6}];
```

To avoid these problems, we redefine b using Mathematica's Piecewise function as follows:

```
Clear[h]
h[x_]:=Piecewise[{{x^2+x,x<0},{1-x,0<x<3},{2x^2-15x+25,
3<=x<=5},{15-2x,x>5}}];
```

**FIGURE 3.2**

(a) Plot does not catch the breaks in the piecewise defined function. (b) If you use Piecewise, Plot can catch jumps.

```
p2 = Plot[h[x], {x, -2, 0, 3, 5, 6}, PlotRange -> All];  
Show[GraphicsRow[{p1, p2}]]
```

Notice that when we execute the Plot command, Mathematica “catches” the breaks between (0,0) and (0,1) and then (5,0) and (5,5) shown in Figure 3.2(b).

From Figure 3.2, we see that $\lim_{x \rightarrow 0} b(x)$ does not exist, $\lim_{x \rightarrow 3} b(x) = -2$, and $\lim_{x \rightarrow 5} b(x)$ does not exist.

When limits exist, you can often use Limit[f[x], x->a] (where a may be $\pm\infty$) to compute $\lim_{x \rightarrow a} f(x)$. Thus, for the previous example we see that

```
Limit[h[x], x -> 3]  
-2
```

is correct. On the other hand,

```
Limit[h[x], x -> 5]  
5
```

is incorrect. We check by computing the right-hand limit, $\lim_{x \rightarrow 5^+} b(x)$, using the Direction->-1 option in the Limit command and then the left limit, $\lim_{x \rightarrow 5^-} b(x)$, using the Direction->1 in the Limit command.

```
Limit[h[x], x -> 5, Direction -> 1]  
0
```

```
Limit[h[x], x -> 5, Direction -> -1]  
5
```

We follow the same procedure for $x = 0$

```
Limit[h[x], x -> 0]  
1
```

Limit[h[x], x → 0, Direction → 1]

0

Limit[h[x], x → 0, Direction → -1]

1

3.1.2 Computing Limits

Some limits involving rational functions can be computed by factoring the numerator and denominator.

Example 3.1.3 Compute $\lim_{x \rightarrow -9/2} \frac{2x^2 + 25x + 72}{72 - 47x - 14x^2}$.

Solution We define `frac1` to be the rational expression $\frac{2x^2 + 25x + 72}{72 - 47x - 14x^2}$. We then attempt to compute the value of `frac1` if $x = -9/2$ by using `ReplaceAll (/.)` to evaluate `frac1` if $x = -9/2$ but see that it is undefined.

```
frac1 /. x -> -9/2
Power::infy: Infinite expression 1/0 encountered. >>
:::indet: Indeterminate expression 0 ComplexInfinity encountered. >>
Indeterminate
```

Factoring the numerator and denominator with `Factor`, `Numerator`, and `Denominator`, we see that

$$\lim_{x \rightarrow -9/2} \frac{2x^2 + 25x + 72}{72 - 47x - 14x^2} = \lim_{x \rightarrow -9/2} \frac{(x+8)(2x+9)}{(8-7x)(2x+9)} = \lim_{x \rightarrow -9/2} \frac{x+8}{8-7x}.$$

The fraction $(x+8)/(8-7x)$ is named `frac2` and the limit is evaluated by computing the value of `frac2` if $x = -9/2$,

Factor[Numerator[frac1]]

Factor[Denominator[frac1]]

$(8+x)(9+2x)$

$-(9+2x)(-8+7x)$

frac2 = Cancel[frac1]

$\frac{-8-x}{-8+7x}$

frac2 /. x -> -9/2

$\frac{7}{79}$

or by using the `Limit` function on the original fraction.

Limit[frac1, x -> -9/2]

$\frac{7}{79}$

We conclude that

$$\lim_{x \rightarrow -9/2} \frac{2x^2 + 25x + 72}{72 - 47x - 14x^2} = \frac{7}{79}.$$

As stated previously, `Limit[f[x], x->a]` attempts to compute $\lim_{x \rightarrow a} f(x)$, `Limit[f[x], x->a, Direction->1]` attempts to compute $\lim_{x \rightarrow a^+} f(x)$, and `Limit[f[x], x->a, Direction->-1]` attempts to compute $\lim_{x \rightarrow a^-} f(x)$. Generally, a can be a number, $\pm\text{Infinity}$ ($\pm\infty$), or another symbol.

Thus, entering

$$\text{Limit}\left[\frac{3x^2 - 7x - 20}{21x^2 + 14x - 35}, x \rightarrow -\frac{5}{3}\right]$$

$\frac{17}{56}$

computes $\lim_{x \rightarrow -9/2} \frac{2x^2 + 25x + 72}{72 - 47x - 14x^2} = 7/79$.

Example 3.1.4 Calculate each limit: (a) $\lim_{x \rightarrow -5/3} \frac{3x^2 - 7x - 20}{21x^2 + 14x - 35}$; (b) $\lim_{x \rightarrow 0} \frac{\sin x}{x}$; (c) $\lim_{x \rightarrow \infty} \left(1 + \frac{z}{x}\right)^x$; (d) $\lim_{x \rightarrow 0} \frac{e^{3x} - 1}{x}$; (e) $\lim_{x \rightarrow \infty} e^{-2x} \sqrt{x}$; and (f) $\lim_{x \rightarrow 1^+} \left(\frac{1}{\ln x} - \frac{1}{x - 1}\right)$.

Solution In each case, we use `Limit` to evaluate the indicated limit. Entering

$$\text{Limit}\left[\frac{3x^2 - 7x - 20}{21x^2 + 14x - 35}, x \rightarrow -\frac{5}{3}\right]$$

$\frac{17}{56}$

computes $\lim_{x \rightarrow -5/3} \frac{3x^2 - 7x - 20}{21x^2 + 14x - 35} = \frac{17}{56}$, and entering

$$\text{Limit}\left[\frac{\sin[x]}{x}, x \rightarrow 0\right]$$

1

computes $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$. Mathematica represents ∞ by `Infinity`. Thus, entering

$$\text{Limit}[(1 + z/x)^x, x \rightarrow \text{Infinity}]$$

e^z

computes $\lim_{x \rightarrow \infty} \left(1 + \frac{z}{x}\right)^x = e^z$. Entering

$$\text{Limit}[(\text{Exp}[3x] - 1)/x, x \rightarrow 0]$$

3

computes $\lim_{x \rightarrow 0} \frac{e^{3x} - 1}{x} = 3$. Entering

$$\text{Limit}[\text{Exp}[-2x]\text{Sqrt}[x], x \rightarrow \text{Infinity}]$$

0

computes $\lim_{x \rightarrow \infty} e^{-2x} \sqrt{x} = 0$, and entering

Because $\ln x$ is undefined for $x \leq 0$, a right-hand limit is mathematically necessary, even though Mathematica's Limit function computes the limit correctly without the distinction.

Limit[1/Log[x] - 1/(x - 1), x → 1, Direction → -1]

$\frac{1}{2}$

computes $\lim_{x \rightarrow 1^+} \left(\frac{1}{\ln x} - \frac{1}{x-1} \right) = \frac{1}{2}$.

3.1.3 One-Sided Limits

As illustrated previously, Mathematica can compute certain one-sided limits. The command **Limit[f[x], x->a, Direction->1]** attempts to compute $\lim_{x \rightarrow a^-} f(x)$, where as **Limit[f[x], x->a, Direction->-1]** attempts to compute $\lim_{x \rightarrow a^+} f(x)$.

Example 3.1.5 Compute (a) $\lim_{x \rightarrow 0^+} |x|/x$; (b) $\lim_{x \rightarrow 0^-} |x|/x$; (c) $\lim_{x \rightarrow 0^+} e^{-1/x}$; and (d) $\lim_{x \rightarrow 0^-} e^{-1/x}$.

Solution Even though $\lim_{x \rightarrow 0} |x|/x$ does not exist, $\lim_{x \rightarrow 0^+} |x|/x = 1$ and $\lim_{x \rightarrow 0^-} |x|/x = -1$, as we see using Limit together with the Direction->1 and Direction->-1 options, respectively.

Limit $\left[\frac{\text{Abs}[x]}{x}, x \rightarrow 0, \text{Direction} \rightarrow 1 \right]$

Limit $\left[\frac{\text{Abs}[x]}{x}, x \rightarrow 0, \text{Direction} \rightarrow -1 \right]$

-1

1

The Direction->-1 and Direction->1 options are used to calculate the correct values for (c) and (d), respectively. For (c), we have

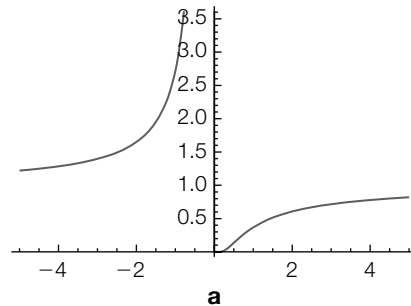
Limit $\left[\frac{1}{x}, x \rightarrow 0, \text{Direction} \rightarrow -1 \right]$

∞

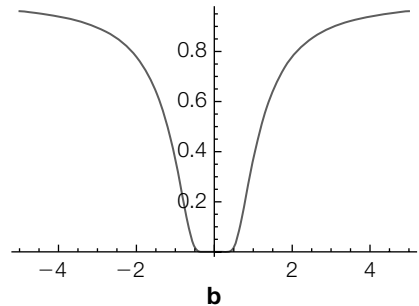
Technically, $\lim_{x \rightarrow 0} e^{-1/x}$ does not exist (see Figure 3.3(a)), so the following is incorrect.

Limit[Exp[-1/x], x → 0]

0



a



b

FIGURE 3.3

(a) Graph of $y = e^{-1/x}$. (b) Graph of $y = e^{-1/x^2}$

However, using Limit together with the Direction option gives the correct left and right limits.

```
Limit[Exp[-1/x], x → 0, Direction → 1]
∞
Limit[Exp[-1/x], x → 0, Direction → -1]
0
```

We confirm these results by graphing $y = e^{-1/x}$ with Plot in Figure 3.3(a). In (b), we also show the graph of $y = e^{-1/x^2}$ in Figure 3.3(b), which is discussed in the exercises.

```
p1 = Plot[Exp[-1/x], {x, -5, 5}];
p2 = Plot[Exp[-1/x^2], {x, -5, 5}];
Show[GraphicsRow[{p1, p2}]]
```

The Limit command together and its options (Direction->1 and Direction->-1) are “fragile” and should be used with caution because the results are unpredictable. It is wise to check or confirm results using a different technique for nearly all problems encountered.

3.1.4 Continuity

Definition 1. The function $y = f(x)$ is **continuous** at $x = a$ if

1. $\lim_{x \rightarrow a} f(x)$ exists;
2. $f(a)$ exists; and
3. $\lim_{x \rightarrow a} f(x) = f(a)$.

Be careful with regard to this. For example, since $\lim_{x \rightarrow 0^-} \sqrt{x}$ does not exist, many would say that $f(x) = \sqrt{x}$ is *right continuous* at $x = 0$.

Note that the third item in the definition means that both (1) and (2) are satisfied. However, if either (1) or (2) is not satisfied, the function is not continuous at the number in question. The function $y = f(x)$ is **continuous** on the open interval I if $f(x)$ is continuous at each number a contained in the interval I . Loosely speaking, the “standard” set of functions (polynomials, rational, trigonometric, etc.) are continuous on their domains.

Example 3.1.6 For what value(s) of x , if any, are each of the following functions continuous? (a) $f(x) = x^3 - 8x$; (b) $f(x) = \sin 2x$; (c) $f(x) = (x - 1)/(x + 1)$; and (d) $f(x) = \sqrt{(x - 1)/(x + 1)}$.

Solution (a) Polynomial functions are continuous for all real numbers. In interval notation, $f(x)$ is continuous on $(-\infty, \infty)$. (b) Because the sine function is continuous for all real numbers, $f(x) = \sin 2x$ is continuous for all real numbers. In interval notation, $f(x)$ is continuous on $(-\infty, \infty)$. (c) The rational function $f(x) = (x - 1)/(x + 1)$ is continuous for all $x \neq -1$. In interval notation, $f(x)$ is continuous on $(-\infty, -1) \cup (-1, \infty)$. (d) $f(x) = \sqrt{(x - 1)/(x + 1)}$ is continuous if the radicand is nonnegative. In

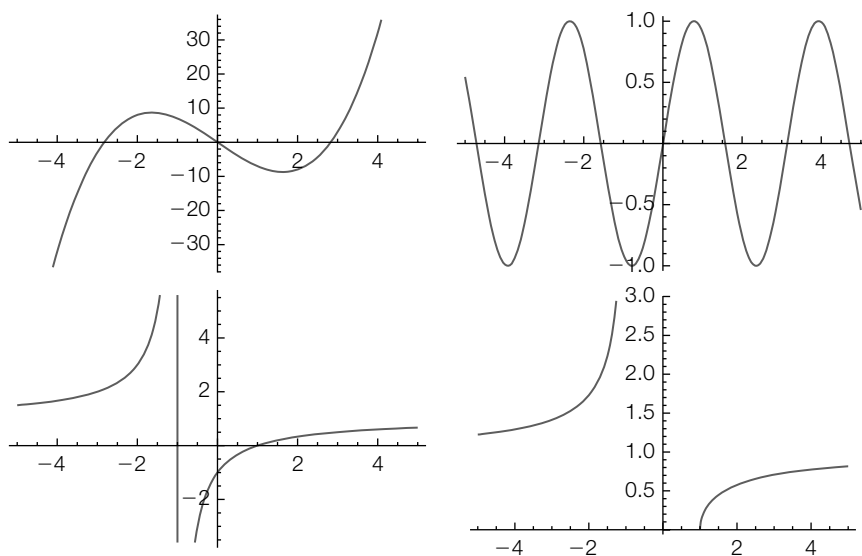


FIGURE 3.4

Polynomials, trigonometric, rational, and root functions are usually continuous on their domains

interval notation, $f(x)$ is strictly continuous on $(-\infty, -1) \cup (1, \infty)$ but some might say that $f(x)$ is continuous on $(-\infty, -1) \cup [1, \infty)$, where it is understood that $f(x)$ is *right continuous* at $x = 1$. We see this by graphing each function with the following commands. See Figure 3.4. Note that in `p3`, the vertical line is *not* a part of the graph of the function—it is a vertical asymptote. If you were to redraw the figure by hand, the vertical line would *not* be a part of the graph.

```
p1 = Plot[x^3 - 8x, {x, -5, 5};
p2 = Plot[Sin[2x], {x, -5, 5};
p3 = Plot[((x - 1)/(x + 1)), {x, -5, 5};
p4 = Plot[Sqrt[(x - 1)/(x + 1)], {x, -5, 5};
Show[GraphicsGrid[{{p1, p2}, {p3, p4}}]]
```

Computers are finite state machines, so handling “interesting” functions can be problematic, especially when one must distinguish between rational and irrational numbers. We assume that if $x = p/q$ is a *rational* number (p and q integers), p/q is a reduced fraction. One way of tackling these sorts of problems is to view rational numbers as ordered pairs, $\{a, b\}$. If a and b are integers, Mathematica automatically reduces a/b so `Denominator[a/b]` or `a/b//Denominator` returns the denominator of the reduced fraction; `Numerator[a/b]` or `a/b//Numerator` returns the numerator of the reduced fraction. If you want to see the points $(x, f(x))$ for which x is rational, we use `ListPlot`.

Example 3.1.7 Let $f(x) = \begin{cases} 1/q, & \text{if } x = p/q \text{ is rational} \\ 0, & \text{if } x \text{ is irrational} \end{cases}$.

Create a representative graph of $f(x)$.

Solution You cannot see points: The measure of the rational numbers is 0, and the measure of the irrational numbers is the **continuum**, $\mathcal{C}$. A true graph of $f(x)$ would look like the graph of $y = 0$. In the context of the example, we want to see how the graph of $f(x)$ looks for rational values of x . We use a few points to illustrate the technique by using **Table** and **Flatten** to generate a set of ordered pairs.

Flatten[list,n] flattens list to level n .

In Mathematica, an ordered pair (a, b) is represented by $\{a, b\}$.

```
t1 = Flatten[Table[{n, m}, {n, 1, 5}, {m, 1, 5}], 1]
{{1, 1}, {1, 2}, {1, 3}, {1, 4}, {1, 5}, {2, 1}, {2, 2},
 {2, 3}, {2, 4}, {2, 5}, {3, 1}, {3, 2}, {3, 3}, {3, 4}, {3, 5}, {4, 1},
 {4, 2}, {4, 3}, {4, 4}, {4, 5}, {5, 1},
 {5, 2}, {5, 3}, {5, 4}, {5, 5}}
```

Next, we defined a function f . Assuming that a and b are integers, given an ordered pair $\{a, b\}$, $f(\{a, b\})$ returns the point $\{a/b, 1/(Reduced\ denominator\ of\ a/b)\}$

```
f[{a_, b_}] := {a/b, 1/(a/b//Denominator)}
```

We use **Map** to compute the value of f for each ordered pair in **t1**. The resulting list is named **t2**.

```
t2 = Map[f, t1]
{{1, 1}, {1/2, 2}, {1/3, 3}, {1/4, 4}, {1/5, 5}, {2, 1}, {1, 1},
 {2/3, 3/2}, {1/2, 2}, {2/5, 5/2}, {3, 1}, {3/2, 2}, {1, 1},
 {3/4, 4/3}, {3/5, 5/3}, {4, 1}, {2, 1}, {4/3, 3/4}, {1, 1},
 {4/5, 5/4}, {5, 1}, {5/2, 2}, {5/3, 3}, {5/4, 4}, {1, 1}}
```

Notice that **t2** contains duplicate entries. We can remove them using **Flatten**, but doing so does not affect the plot shown in Figure 3.5(a).

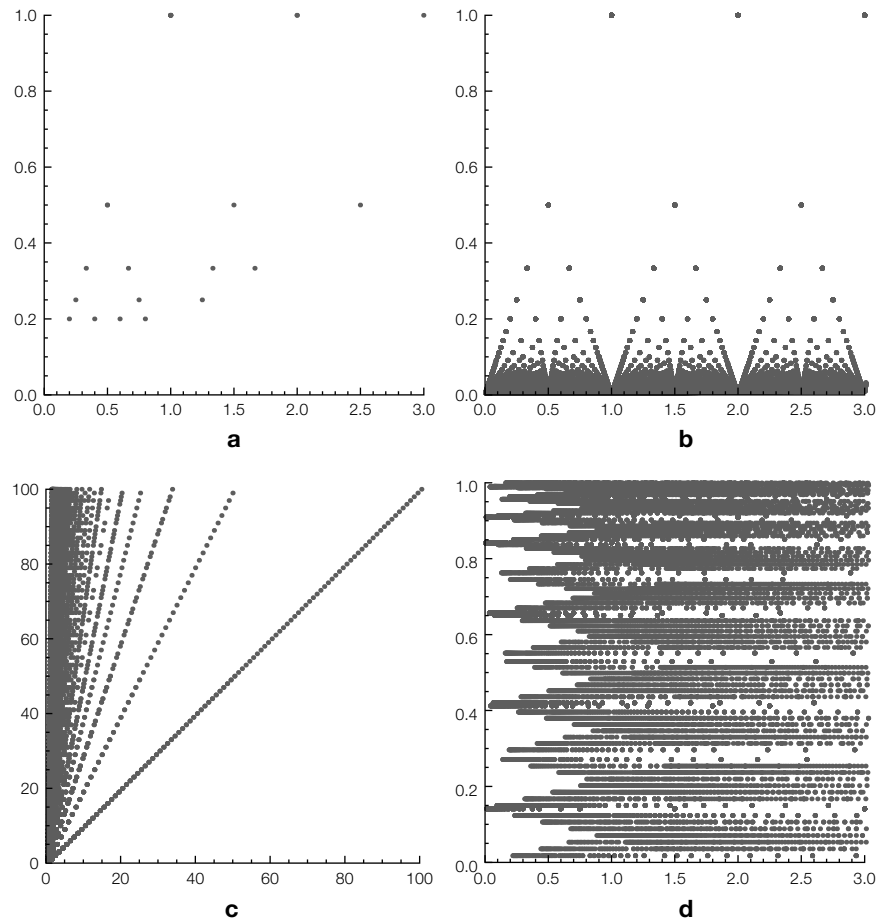
```
p1 = ListPlot[t2, PlotRange → {{0, 3}, {0, 1}}, AspectRatio → 1];
```

To generate a “prettier” plot, we repeat the procedure using more points. After entering each command, the results are not displayed because we include a semicolon (;) at the end of each. See Figure 3.5(b).

```
t3 = Flatten[Table[{n, m}, {n, 1, 300}, {m, 1, 200}], 1];
t4 = Map[f, t3];
p2 = ListPlot[t4, PlotRange → {{0, 3}, {0, 1}}, AspectRatio → 1];
```

This function is interesting because it is continuous at the irrationals and discontinuous at the rationals.

We can consider other functions in similar contexts. In the following, the y -coordinate is the numerator rather than the denominator. See Figure 3.5(c).

**FIGURE 3.5**

(a) After step 1. (b) After step 2. (c) Examining the numerator rather than the denominator. (d) The sine of the numerator

```
Clear[f]
f[{a_, b_}] := {a/b, a/b//Numerator};
t3 = Flatten[Table[{n, m}, {n, 1, 100}, {m, 1, 100}], 1];
t4 = Map[f, t3];
p3 = ListPlot[t4, PlotRange -> {{0, 100}, {0, 100}}, AspectRatio -> 1];
```

With Mathematica, we can modify commands to investigate how changing parameters affect a given situation. In the following, we compute the sine of p if $x = p/q$. See Figure 3.5(d).

```
Clear[f]
f[{a_, b_}] := {a/b, Sin[(a/b//Numerator)]};
```

```

t5 = Flatten[Table[{n, m}, {n, 1, 300}, {m, 1, 200}], 1];
t6 = Map[f, t5];
p4 = ListPlot[t6, PlotRange → {{0, 3}, {0, 1}}, AspectRatio → 1];
Show[GraphicsGrid[{{p1, p2}, {p3, p4}}]]

```

3.2 DIFFERENTIAL CALCULUS

3.2.1 Definition of the Derivative

Definition 2. *The derivative of $y = f(x)$ is*

$$y' = f'(x) = \frac{dy}{dx} = \lim_{b \rightarrow 0} \frac{f(x+b) - f(x)}{b}, \quad (3.1)$$

provided the limit exists.

Assuming that $(a, f(a))$ and $(a+b, f(a+b))$ exist, the line with equation $y = \frac{f(a+b) - f(a)}{b}(x - a) + f(a)$ is the secant containing the two points. Assuming the derivative exists, as b approaches 0, the secants approach the tangent. Hence, if the limit exists, the derivative gives us the slope of a function at that particular value of x .

The Limit command can be used along with Simplify to compute the derivative of a function using the definition of the derivative.

Example 3.2.1 Use the definition of the derivative to compute the derivative of (a) $f(x) = x + 1/\sqrt{x}$ and (b) $g(x) = 1/\sqrt{x}$.

Solution For (a), we first define f , compute the difference quotient, $(f(x+h) - f(x))/h$, simplify the difference quotient with Simplify, and use Limit to calculate the derivative.

```

f[x_] = x + 1/x;
step1 = (f[x + h] - f[x])/h
step2 = Simplify[step1]
Limit[step2, h → 0]

```

$$\begin{aligned}
 & \frac{h - \frac{1}{x} + \frac{1}{h+x}}{h} \\
 & \frac{-1 + hx + x^2}{x(h+x)} \\
 & 1 - \frac{1}{x^2}
 \end{aligned}$$

For (b), we use the same approach as in (a) but use Together rather than Simplify to reduce the complex fraction.

```

step1 = (g[x + h] - g[x])/h
step2 = Together[step1]

```

Limit[step2, h → 0]

$$\frac{-\frac{1}{\sqrt{x}} + \frac{1}{\sqrt{h+x}}}{\frac{h}{\sqrt{x} - \sqrt{h+x}}}$$

$$-\frac{1}{2x^{3/2}}$$

If the derivative of $y = f(x)$ exists at $x = a$, a geometric interpretation of $f'(a)$ is that $f'(a)$ is the slope of the line tangent to the graph of $y = f(x)$ at the point $(a, f(a))$.

To motivate the definition of the derivative, many calculus texts choose a value of x , $x = a$, and then draw the graph of the secant line passing through the points $(a, f(a))$ and $(a + b, f(a + b))$ for “small” values of b to show that as b approaches 0, the secant line approaches the tangent line. An equation of the secant line passing through the points $(a, f(a))$ and $(a + b, f(a + b))$ is given by

$$y - f(a) = \frac{f(a + b) - f(a)}{(a + b) - a} (x - a) \quad \text{or} \quad y = \frac{f(a + b) - f(a)}{b} (x - a) + f(a).$$

Example 3.2.2 If $f(x) = x^2 - 4x$, graph $f(x)$ together with the secant line containing $(1, f(1))$ and $(1 + b, f(1 + b))$ for various values of b .

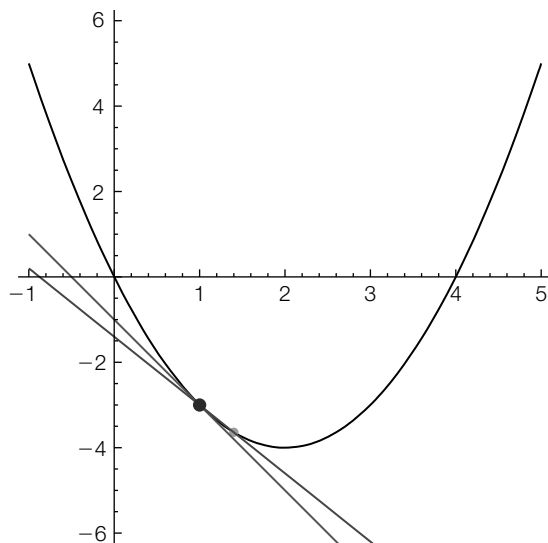
Solution We begin by considering a particular b value. We choose $b = 0.4$. We then define $f(x) = x^2 - 4x$. In p1, we graph $f(x)$ in black on the interval $[-1, 5]$, in p2 we place a blue point at $(1, f(1))$ and a green point at $(1.4, f(1.4))$, in p3 we graph the tangent to $y = f(x)$ at $(1, f(1))$ in red, in p4 we graph the secant containing $(1, f(1))$ and $(1.4, f(1.4))$ in purple, and finally we show all four graphics together with Show in Figure 3.6.

Remember that when a semicolon is placed at the end of a command, the resulting output is not displayed. The names of the colors that Mathematica knows are listed in the **ColorSchemes** palette followed by “Known” and then “System.”

```
f[x_] = x^2 - 4x;
p1 = Plot[f[x], {x, -1, 5}, PlotStyle → Black];
p2 = Graphics[{PointSize[.03], Blue, Point[{1, f[1]}],
  Green, Point[{1 + .4, f[1 + .4]}]}];
p3 = Plot[f'[1](x - 1) + f[1], {x, -1, 5}, PlotStyle → Red];
p4 = Plot[(f[1 + .4] - f[1])/.4(x - 1) + f[1], {x, -1, 5},
  PlotStyle → Purple];
Show[p1, p2, p3, p4, PlotRange → {{-1, 5}, {-6, 6}},
  AspectRatio → 1]]
```

We now generalize the previous set of commands for arbitrary $b \neq 0$ values. $g(b)$ shows plots of $y = x^2 - 4x$, the tangent at $(1, f(1))$, and the secant containing $(1, f(1))$ and $(1 + b, f(1 + b))$.

```
Clear[f, g];
f[x_] = x^2 - 4x;
g[h_] := Module[{p1, p2, p3, p4},
  p1 = Plot[f[x], {x, -1, 5}, PlotStyle → Black];
```

**FIGURE 3.6**

Plots of $y = x^2 - 4x$, the tangent at $(1, f(1))$, and the secant containing $(1, f(1))$ and $(1 + h, f(1 + h))$ if $h = 0.4$

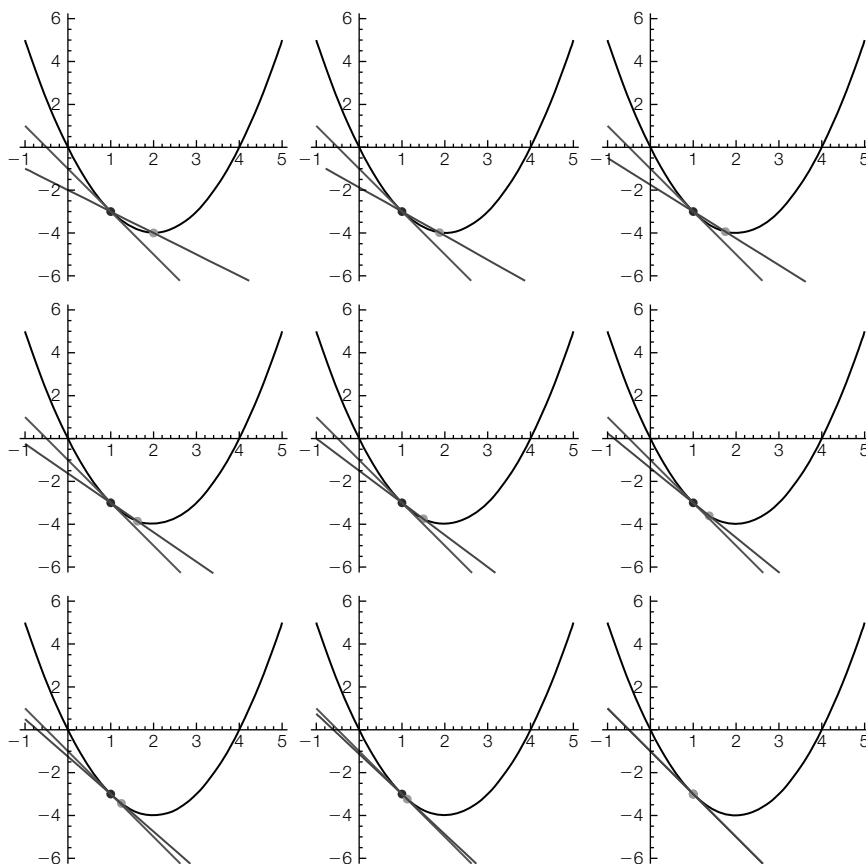
```
p2 = Graphics[{PointSize[.03], Blue, Point[{1, f[1]}],
  Green, Point[{1 + h, f[1 + h]}]}];
p3 = Plot[f'[1](x - 1) + f[1], {x, -1, 5}, PlotStyle → Red];
p4 = Plot[(f[1 + h] - f[1])/h(x - 1) + f[1], {x, -1, 5},
  PlotStyle → Purple];
Show[p1, p2, p3, p4, PlotRange → {{-1, 5}, {-6, 6}}, AspectRatio → 1]]
```

`Table[f[x], {x, start, stop, stepsize}]` creates a table of $f(x)$ values beginning with *start* and ending with *stop* using increments of *stepsize*. Given a table, `Partition[table, n]` partitions the table into n element subgroups. Thus, if a table, `t1`, has nine elements, `Partition[t1, 3]` creates a 3×3 grid; three sets of three elements each.

Using `Table` followed by `GraphicsGrid`, we can create a table of graphics for various values of h like that shown in Figure 3.7. With `Table`, the dimensions of the grid displayed on your computer are based on the size of the active Mathematica window. To control the dimensions of the grid, we use `GraphicsGrid` together with `Partition` and `Show`.

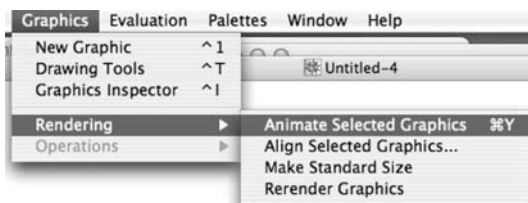
```
t1 = Table[g[k], {k, 1, .0001, -(1 - .0001)/8}]
Show[GraphicsGrid[Partition[t1, 3]]]
```

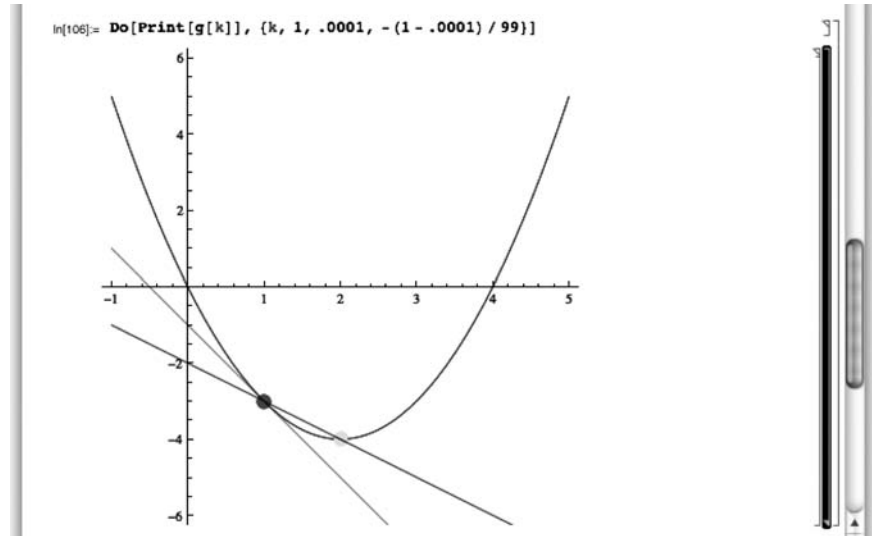
`Do` works in the same way as `Table`. Rather than creating a table (or list), `Do` performs the action repeatedly. Thus, you can use `Do` to create an animation of the secants approaching the tangent.

**FIGURE 3.7**

Plots of $y = x^2 - 4x$, the tangent at $(1, f(1))$, and the secant containing $(1, f(1))$ and $(1+b, f(1+b))$ for various values of b

Thus, entering computes $g(k)$ and displays the result for 100 equally spaced values of k starting with $k = 1$ and ending with $k = 0.0001$. To animate the results, select the graphics as indicated. Go to the Mathematica menu, select **Graphics** followed by **Rendering** and then **Animate Selected Graphics**





Keyboard shortcuts are usually operating system dependent.

or use a keyboard shortcut to animate the graphics. After animating the selection, you can control the animation (speed, direction, and so on) with the buttons  displayed in the lower left-hand corner of the Mathematica notebook.

With Mathematica 6, you can use **Manipulate** to help generate animations and images that you can adjust based on changing parameter values.

To illustrate how to do so, we begin by redefining f and then defining $m(a, b)$. Given a and b values, $m(a, b)$ plots $f(x)$ for $-10 \leq x \leq 10$ (p1), plots a blue point $(a, f(a))$ and a green point at $(a + b, f(a + b))$ (p2), plots $f'(a)(x - a) + f(a)$ (the tangent to the graph of $f(x)$ at $(a, f(a))$) for $-1 \leq x \leq 5$ in red (p3), the secant containing $(a, f(a))$ and $(a + b, f(a + b))$ for $-10 \leq x \leq 10$ in purple (p4), and finally displays all four graphics together with **Show**. Using **PlotRange**, we indicate that the horizontal axis displays x values between -10 and 10 , at the vertical axis displays y values between -10 and 10 ; **AspectRatio** $\rightarrow 1$ means that the ratio of the lengths of the x to y axes is 1. Thus, the plot scaling is correct. Note that when we use **Module** to define m , p1, p2, p3, and p4 are *local* to the function m . This means that if you have such objects defined elsewhere in your Mathematica notebook, those objects are not affected when you compute m .

```
Clear[m, f];
f[x_] = x^2 - 4x;
m[a_, h_] := Module[{p1, p2, p3, p4},
  p1 = Plot[f[x], {x, -10, 10}, PlotStyle -> Black];
  p2 = Graphics[{PointSize[.03], Blue, Point[{a, f[a]}],
    Green, Point[{a + h, f[a + h]}]}];
  p3 = Plot[f'[a](x - a) + f[a], {x, -1, 5}, PlotStyle -> Red];
```

```

p4 = Plot[(f[a + h] - f[a])/h(x - a) + f[a], {x, -10, 10},
PlotStyle -> Purple];
Show[p1, p2, p3, p4, PlotRange -> {{-10, 10}, {-10, 10}},
AspectRatio -> 1]]

```

Now we use `Manipulate` to create a “mini” program. The sliders (centered at $a = 0$ and $b = 0.5$ with a range from -10 to 10 and -1 to 1 , respectively) allow you to see how changing a and b affects the plot. See Figure 3.8.

```

Manipulate[m[a, h], {{a, 0}, -10, 10}, {{h, .5}, -1, 1]]

```

Figure 3.8 illustrates the special case in which $f(x) = x^2 - 4x$. To illustrate the same concept using a “standard” set of functions (polynomials, rational, root, and trig), we first define the functions

```

quad[x_] = (x + 2)^2 - 2;
cubic[x_] = -1/10x(x^2 - 25);
rational[x_] = 50/((x + 5)(x - 5));
root[x_] = 3Sqrt[x + 5];
sin[x_] = 5Sin[x];

```

and then we adjust m by defining a few of these “standard” and then defining the function `mmore`. `mmore` performs the same actions as m but does so for the

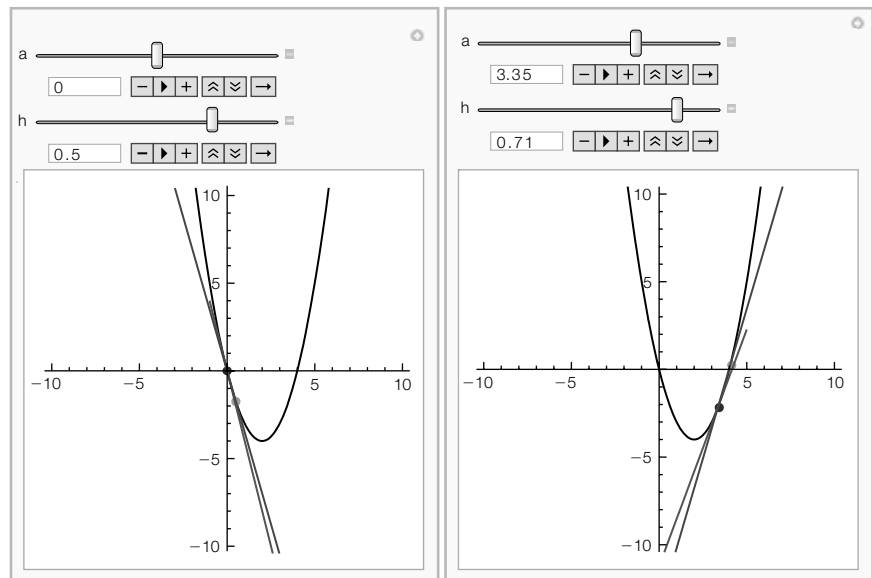


FIGURE 3.8

With `Manipulate`, we can perform animations and see how a function changes depending on parameter values

function selected. We then use `Manipulate` to create an object that shows the secant (in purple), the tangent (in red) for the selected function, a value, and b value. See Figure 3.9.

```
Clear[mmore];
mmore[f_, a_, h_] := Module[{p1, p2, p3, p4},
  p1 = Plot[f[x], {x, -10, 10}, PlotStyle → Black];
  p2 = Graphics[{PointSize[.03], Blue, Point[{a, f[a]}],
    Green, Point[{a + h, f[a + h]}]}];
  p3 = Plot[f'[a](x - a) + f[a], {x, -10, 10}, PlotStyle → Red];
  p4 = Plot[(f[a + h] - f[a])/h(x - a) + f[a], {x, -10, 10},
    PlotStyle → Purple];
  Show[p1, p2, p3, p4, PlotRange → {{-10, 10}, {-10, 10}},
    AspectRatio → 1]]

Manipulate[mmore[f, a, h], {{f, quad},
  {quad, cubic, rational, root, sin}},
  {{a, 0}, -10, 10}, {{h, 1}, -2, 2}]
```

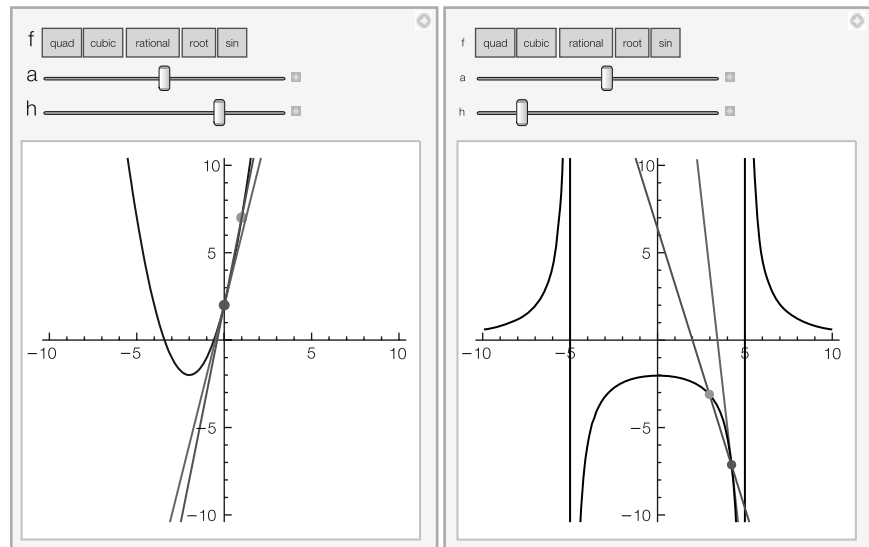


FIGURE 3.9

With this `Manipulate` object, we see how various functions, a values, and b values affect the secant to $y = f(x)$ passing through $(a, f(a))$ and $(a + b, f(a + b))$ and the tangent to $y = f(x)$ at $(a, f(a))$

3.2.2 Calculating Derivatives

The functions **D** and **'** are used to differentiate functions. Assuming that $y = f(x)$ is differentiable,

1. **D[f[x],x]** computes and returns $f'(x) = df/dx$,
2. **f'[x]** computes and returns $f'(x) = df/dx$,
3. **f''[x]** computes and returns $f^{(2)}(x) = d^2f(x)/dx^2$, and
4. **D[f[x],x,n]** computes and returns $f^{(n)}(x) = d^n f(x)/dx^n$.
5. You can use the  button located on the **BasicMathInput** palette to create templates to compute derivatives.

Figure 3.10 illustrates various ways of computing derivatives using the **'** symbol, **D**, and the ∂ symbol.

Mathematica knows the numerous differentiation rules, including the product, quotient, and chain rules. Thus, entering

```
Clear[f, g]
D[f[x]g[x], x]
g[x]f'[x] + f[x]g'[x]
```

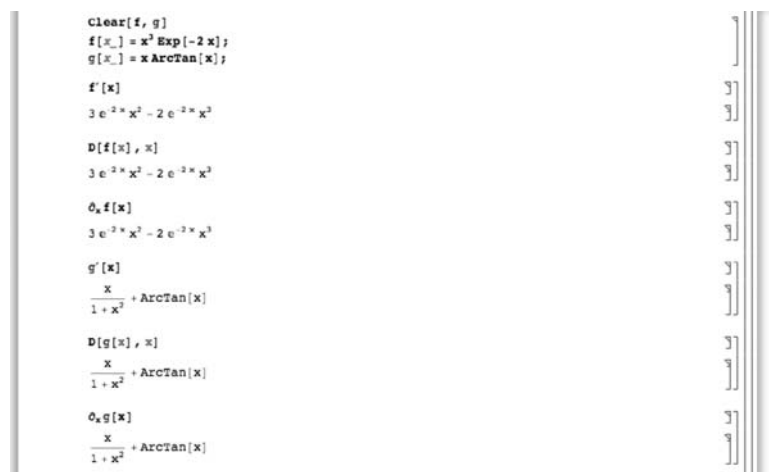
shows us that $\frac{d}{dx}(f(x) \cdot g(x)) = f'(x)g(x) + f(x)g'(x)$; entering

```
Together[D[f[x]/g[x], x]]
g[x]f'[x] - f[x]g'[x]
g[x]^2
```

```
D[f[g[x]], x]
```

shows us that $\frac{d}{dx}(f(x)/g(x)) = (f'(x)g(x) - f(x)g'(x))/(g(x))^2$; and entering

Throughout the text,
input is in bold and
output is not; output
follows input.



```
Clear[f, g]
f[x_] = x^3 Exp[-2 x];
g[x_] = x ArcTan[x];

f'[x]
3 e-2 x x2 - 2 e-2 x x3

D[f[x], x]
3 e-2 x x2 - 2 e-2 x x3

∂x f[x]
3 e-2 x x2 - 2 e-2 x x3

g'[x]
x / (1 + x2) + ArcTan[x]

D[g[x], x]
x / (1 + x2) + ArcTan[x]

∂x g[x]
x / (1 + x2) + ArcTan[x]
```

FIGURE 3.10

You can use **'**, **D**, and ∂ to compute derivatives of functions

D[f[g[x]], x]

$f'[g[x]]g'[x]$

shows us that $\frac{d}{dx}(f(g(x))) = f'(g(x))g'(x)$.

Example 3.2.3 Compute the first and second derivatives of (a) $y = x^4 + \frac{4}{3}x^3 - 3x^2$, (b) $f(x) = 4x^5 - \frac{5}{2}x^4 - 10x^3$, (c) $y = \sqrt{e^{2x} + e^{-2x}}$, and (d) $y = (1 + 1/x)^x$.

Solution For (a), we use D.

D[x^4 + 4/3x^3 - 3x^2, {x, 2}]
 $-6 + 8x + 12x^2$

For (b), we first define f and then use ' together with Factor to calculate and factor $f'(x)$ and $f''(x)$.

f[x_] = 4x^5 - 5/2x^4 - 10x^3;
Factor[f'[x]]
 $10x^2(1 + x)(-3 + 2x)$
Factor[f''[x]]
 $10x(-6 - 3x + 8x^2)$

For (c), we use Simplify together with D to calculate and simplify y' and y'' .

D[Sqrt[Exp[2x] + Exp[-2x]], {x, 2}]/Simplify

$$\frac{\sqrt{e^{-2x} + e^{2x}}(1 + 6e^{4x} + e^{8x})}{(1 + e^{4x})^2}$$

By hand, (d) would require logarithmic differentiation. The second derivative would be particularly difficult to compute by hand. Mathematica quickly computes and simplifies each derivative.

Simplify[D[(1 + 1/x)^x, x]]

$$\frac{(1 + \frac{1}{x})^x(-1 + (1 + x)\text{Log}[1 + \frac{1}{x}])}{1 + x}$$

Simplify[D[(1 + 1/x)^x, {x, 2}]]

$$\frac{(1 + \frac{1}{x})^x(-1 + x - 2x(1 + x)\text{Log}[1 + \frac{1}{x}] + x(1 + x)^2\text{Log}[1 + \frac{1}{x}]^2)}{x(1 + x)^2}$$

Map and operations on lists are discussed in more detail in Chapter 4.

The command Map[f,list] applies the function f to each element of the list list. Thus, if you are computing the derivatives of a large number of functions, you can use Map together with D.

Remark 3.2 A built-in Mathematica function is **threadable** if f[list] returns the same result as Map[f,list]. Many familiar functions such as D and Integrate are threadable.

Example 3.2.4 Compute the first and second derivatives of $\sin x$, $\cos x$, $\tan x$, $\sin^{-1} x$, $\cos^{-1} x$, and $\tan^{-1} x$.

Solution Notice that lists are contained in braces. Thus, entering

$$\begin{aligned} &\text{Map}[D[\#, x] \&, \{\text{Sin}[x], \text{Cos}[x], \text{Tan}[x], \\ &\quad \text{ArcSin}[x], \text{ArcCos}[x], \text{ArcTan}[x]\}] \\ &\left\{ \text{Cos}[x], -\text{Sin}[x], \text{Sec}[x]^2, \frac{1}{\sqrt{1-x^2}}, -\frac{1}{\sqrt{1-x^2}}, \frac{1}{1+x^2} \right\} \end{aligned}$$

computes the first derivative of the three trigonometric functions and their inverses. In this case, we have applied a *pure function* to the list of trigonometric functions and their inverses. Given an argument $\#$, $D[\#, x] \&$ computes the derivative of $\#$ with respect to x . The $\&$ symbol is used to mark the end of a *pure function*. Similarly, entering

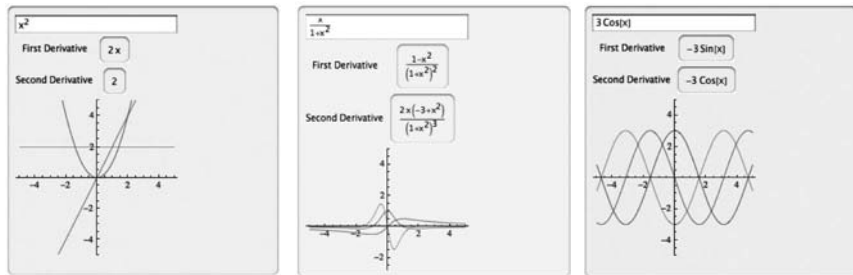
$$\begin{aligned} &\text{Map}[D[\#, \{x, 2\}] \&, \{\text{Sin}[x], \text{Cos}[x], \text{Tan}[x], \\ &\quad \text{ArcSin}[x], \text{ArcCos}[x], \text{ArcTan}[x]\}] \\ &\left\{ -\text{Sin}[x], -\text{Cos}[x], 2\text{Sec}[x]^2 \text{Tan}[x], \frac{x}{(1-x^2)^{3/2}}, -\frac{x}{(1-x^2)^{3/2}}, -\frac{2x}{(1+x^2)^2} \right\} \end{aligned}$$

computes the second derivative of the three trigonometric functions and their inverses. Because D is threadable, the same results are obtained with the following commands:

$$\begin{aligned} &D[\{\{\text{Sin}[x], \text{Cos}[x], \text{Tan}[x], \\ &\quad \text{ArcSin}[x], \text{ArcCos}[x], \text{ArcTan}[x]\}, x] \\ &\left\{ \text{Cos}[x], -\text{Sin}[x], \text{Sec}[x]^2, \frac{1}{\sqrt{1-x^2}}, -\frac{1}{\sqrt{1-x^2}}, \frac{1}{1+x^2} \right\} \\ &D[\{\{\text{Sin}[x], \text{Cos}[x], \text{Tan}[x], \\ &\quad \text{ArcSin}[x], \text{ArcCos}[x], \text{ArcTan}[x]\}, \{x, 2\}] \\ &\left\{ -\text{Sin}[x], -\text{Cos}[x], 2\text{Sec}[x]^2 \text{Tan}[x], \frac{x}{(1-x^2)^{3/2}}, -\frac{x}{(1-x^2)^{3/2}}, -\frac{2x}{(1+x^2)^2} \right\} \end{aligned}$$

With `DynamicModule`, we create a simple dynamic that lets you compute the first and second derivatives of basic functions and plot them on a standard viewing window, $[-5, 5] \times [-5, 5]$. The layout of Figure 3.11 is primarily determined by `Panel`, `Column`, and `Grid`.

```
Panel[DynamicModule[{f = x^2},
  Column[{InputField[Dynamic[f]], Grid[{{"FirstDerivative",
    Panel[Dynamic[D[f, x]/Simplify]]},
    {"SecondDerivative", Panel[Dynamic[D[f, {x, 2}]/Simplify]]}],
  Dynamic[Plot[Evaluate[Tooltip[{f, D[f, x], D[f, {x, 2}]]],
    {x, -5, 5}, PlotRange -> {-5, 5},
    AspectRatio -> Automatic]]], ImageSize -> {300, 300}]
```

**FIGURE 3.11**

Seeing the relationship between the first and second derivative of a function and the original function

3.2.3 Implicit Differentiation

If an equation contains two variables, x and y , implicit differentiation can be carried out by explicitly declaring y to be a function of x , $y = y(x)$, and using D or by using the Dt command.

Example 3.2.5 Find $y' = dy/dx$ if (a) $\cos(e^{xy}) = x$ and (b) $\ln(x/y) + 5xy = 3y$.

Solution For (a) we illustrate the use of D . Notice that we are careful to specifically indicate that $y = y(x)$. First we differentiate with respect to x .

```
Clear[x, y]
s1 = D[Cos[Exp[xy[x]]] - x, x]
-1 - exy[x] Sin[exy[x]] (y[x] + xy'[x])
```

and then we solve the resulting equation for $y' = dy/dx$ with $Solve$.

```
Solve[s1==0, y'[x]]
{{y'[x] -> - (e-xy[x] Csc[exy[x]] (1 + exy[x] Sin[exy[x]] y[x]) / x}}
```

For (b), we use Dt . When using Dt , we interpret $Dt[x] = 1$ and $Dt[y] = y' = dy/dx$. Thus, entering

```
s2 = Dt[Log[x/y] + 5xy - 3y]
5yDt[x] - 3Dt[y] + 5xDt[y] + (y (Dt[x]/y - xDt[y]/y2)) / x
s3 = s2/.{Dt[x] -> 1, Dt[y] -> dydx}
-3dydx + 5dydx + 5y + (- (dydx/x + 1/y))y
```

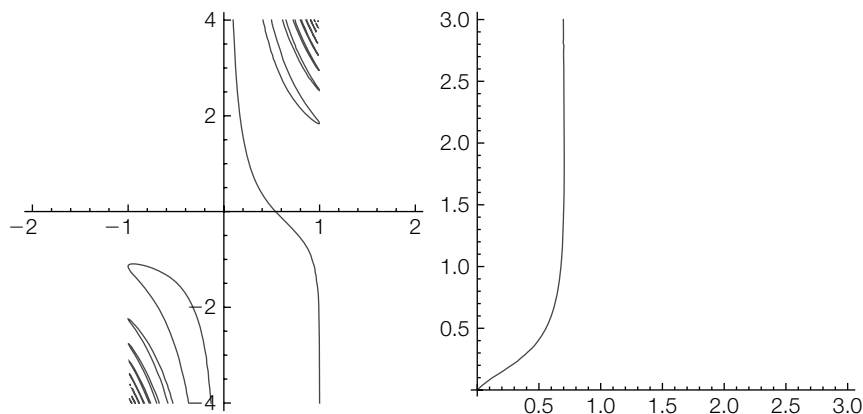


FIGURE 3.12

On the left, $\cos(e^{xy}) = x$ for $-2 \leq x \leq 2$ and $-4 \leq y \leq 4$; on the right, $\ln(x/y) + 5xy = 3y$ for $0.01 \leq x \leq 3$ and $0.01 \leq y \leq 3$.

and solving for dy/dx with `Solve`

```
Solve[s3==0, dydx]
{{ dydx -> - (y(1 + 5xy)) / (x(-1 - 3y + 5xy)) }}
```

shows us that if $\ln(x/y) + 5xy = 3y$, $y' = \frac{dy}{dx} = -\frac{(1 + 5xy)y}{(5xy - 3y - 1)x}$.

To graph each equation, we use `ContourPlot`. Generally, given an equation of the form $f(x, y) = g(x, y)$, the command

```
ContourPlot[f[x, y]==g[x, y], {x, a, b}, {y, c, d}]
```

attempts to plot the graph of $f(x, y) = g(x, y)$ on the rectangle $[a, b] \times [c, d]$. Using `Show` together with `GraphicsRow`, we show the two graphs side-by-side in Figure 3.12.

```
cp1 = ContourPlot[Cos[Exp[xy]]==x, {x, -2, 2}, {y, -4, 4}, PlotPoints -> 120,  
Frame -> False, Axes -> Automatic, AxesOrigin -> {0, 0}];  
cp2 = ContourPlot[Log[x/y] + 5xy==3y, {x, .01, 3}, {y, .01, 3},  
PlotPoints -> 120,  
Frame -> False, Axes -> Automatic, AxesOrigin -> {0, 0}];  
Show[GraphicsRow[{cp1, cp2}]]
```

3.2.4 Tangent Lines

If $f'(a)$ exists, a typical interpretation of $f'(a)$ is that $f'(a)$ is the slope of the line tangent to the graph of $y = f(x)$ at the point $(a, f(a))$. In this case, an equation of the tangent is given by

$$y - f(a) = f'(a)(x - a) \quad \text{or} \quad y = f'(a)(x - a) + f(a).$$

Example 3.2.6 Find an equation of the line tangent to the graph of $f(x) = \sin x^{1/3} + \cos^{1/3} x$ at the point with x -coordinate $x = 5\pi/3$.

Solution Because we will be graphing a function involving odd roots of negative numbers, we begin by loading the **RealOnly** package contained in the **Miscellaneous** folder (or directory). We then define $f(x)$ and compute $f'(x)$.

If this package was not included with your version of Mathematica, you may want to download it from the Wolfram website.

```
<< Miscellaneous`RealOnly`
f[x_] = Sin[x^(1/3)] + Cos[x]^(1/3);
f'[x] =
  Cos[x^(1/3)]
  -----
  3x^(2/3)
  -
  Sin[x]
  -----
  3Cos[x]^(2/3)
```

Then, the slope of the line tangent to the graph of $f(x)$ at the point with x -coordinate $x = 5\pi/3$ is

```
f'[5Pi/3]
  1
  -----
  2^(1/3)√3
  +
  Cos[(5π/3)^(1/3)]
  -----
  3^(1/3)(5π)^(2/3)

f'[5Pi/3]/N
0.440013
```

while the y -coordinate of the point is

```
f[5Pi/3]
  1
  -----
  2^(1/3)
  + Sin[(5π/3)^(1/3)]

f[5Pi/3]/N
1.78001
```

Thus, an equation of the line tangent to the graph of $f(x)$ at the point with x -coordinate $x = 5\pi/3$ is

$$y - \left(\frac{1}{\sqrt[3]{2}} + \sin \sqrt[3]{5\pi/3} \right) = \left(\frac{\cos \sqrt[3]{5\pi/3}}{\sqrt[3]{3}\sqrt[3]{25\pi^2}} + \frac{1}{\sqrt[3]{2}\sqrt[3]{3}} \right) \left(x - \frac{5\pi}{3} \right),$$

as shown in Figure 3.13.

```
p1 = Plot[f[x], {x, 0, 4Pi}, PlotStyle → Black];
p2 = ListPlot[{5Pi/3, f[5Pi/3]/N}, PlotStyle → PointSize[.03];
```

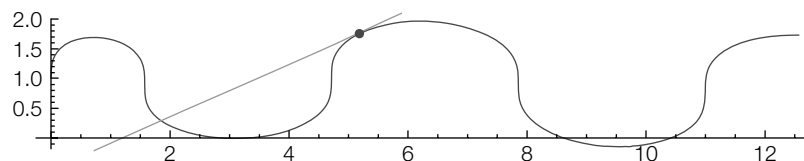


FIGURE 3.13

$f(x) = \sin x^{1/3} + \cos^{1/3} x$ together with its tangent at the point $(5\pi/3, f(5\pi/3))$

```
p3 = Plot[f'[5Pi/3](x - 5Pi/3) + f[5Pi/3], {x, 0, 4Pi},
PlotStyle -> GrayLevel[.6]];
```

```
Show[p1, p2, p3, AspectRatio -> Automatic,
DisplayFunction -> $DisplayFunction]
```

Tangent Lines of Implicit Functions

Example 3.2.7 Find equations of the tangent line and normal line to the graph of $x^2y - y^3 = 8$ at the point $(-3, 1)$. Find and simplify $y'' = d^2y/dx^2$.

Solution We evaluate $y' = dy/dx$ if $x = -3$ and $y = 1$ to determine the slope of the tangent line at the point $(-3, 1)$. Note that we cannot (easily) solve $x^2y - y^3 = 8$ for y , so we use implicit differentiation to find $y' = dy/dx$:

By the product and chain rules, $\frac{d}{dx}(x^2y) = \frac{d}{dx}(x^2)y + x^2\frac{d}{dx}(y) = 2x \cdot y + x^2 \cdot \frac{dy}{dx} = 2xy + x^2y'$.

$$\begin{aligned}\frac{d}{dx}(x^2y - y^3) &= \frac{d}{dx}(8) \\ 2xy + x^2y' - 3y^2y' &= 0 \\ y' &= \frac{-2xy}{x^2 - 3y^2}.\end{aligned}$$

```
eq = x^2y - y^3 == 8
s1 = Dt[eq]
2xyDt[x] + x^2Dt[y] - 3y^2Dt[y] == 0
s2 = s1/.Dt[x] -> 1
2xy + x^2Dt[y] - 3y^2Dt[y] == 0
s3 = Solve[s2, Dt[y]]
{{Dt[y] -> -2xy/(x^2 - 3y^2)}}
```

Lists are discussed in more detail in Chapter 4.

Notice that **s3** is a **list**. The formula for $y' = dy/dx$ is the second part of the first part of the first part of **s3** and extracted from **s3** with

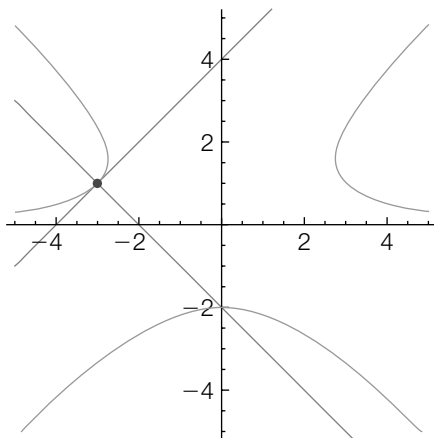
```
s3[[1,1,2]]
-2xy/(x^2 - 3y^2)
```

We then use **ReplaceAll (/.)** to find that the slope of the tangent at $(-3, 1)$ is

```
s3[[1,1,2]]/.{x -> -3, y -> 1}
1
```

The slope of the normal is $-1/1 = -1$. Equations of the tangent and normal are given by

$$y - 1 = 1(x + 3) \quad \text{and} \quad y - 1 = -1(x + 3),$$

**FIGURE 3.14**

Graphs of $x^2y - y^3 = 8$ (in black) and the tangent and normal at $(-3, 1)$ (in gray)

respectively. See Figure 3.14.

```
cp1 = ContourPlot[x^2y - y^3 - 8, {x, -5, 5}, {y, -5, 5}, Contours -> {0},  
ContourShading -> False, PlotPoints -> 200];  
p1 = ListPlot[{{-3, 1}}, PlotStyle -> PointSize[.03]];  
p2 = Plot[{(x + 3) + 1, -(x + 3) + 1}, {x, -5, 5}, PlotStyle -> Gray];  
Show[cp1, p1, p2, Frame -> False, Axes -> Automatic,  
AxesOrigin -> {0, 0}, AspectRatio -> Automatic,  
DisplayFunction -> $DisplayFunction]
```

To find $y'' = d^2y/dx^2$, we proceed as follows:

```
s4 = Dt[s3[[1, 1, 2]]]/Simplify  

$$-\frac{2(x^2 + 3y^2)(-yDt[x] + xDt[y])}{(x^2 - 3y^2)^2}$$
  
s5 = s4.Dt[x] -> 1/.s3[[1]]/Simplify  

$$\frac{6y(x^2 - y^2)(x^2 + 3y^2)}{(x^2 - 3y^2)^3}$$

```

The result means that

$$y'' = \frac{d^2y}{dx^2} = \frac{6(x^2y - y^3)(x^2 + 3y^2)}{(x^2 - 3y^2)^3}.$$

Because $x^2y - y^3 = 8$, the second derivative is further simplified to

$$y'' = \frac{d^2y}{dx^2} = \frac{48(x^2 + 3y^2)}{(x^2 - 3y^2)^3}.$$

Parametric Equations and Polar Coordinates

For the parametric equations $\{x = f(t), y = g(t)\}$, $t \in I$,

$$y' = \frac{dy}{dx} = \frac{dy/dt}{dx/dt} = \frac{g'(t)}{f'(t)}$$

and

$$y'' = \frac{d^2y}{dx^2} = \frac{d}{dx} \frac{dy}{dx} = \frac{d/dt(dy/dx)}{dx/dt}.$$

If $\{x = f(t), y = g(t)\}$ has a tangent line at the point $(f(a), g(a))$, parametric equations of the tangent are given by

$$x = f(a) + tf'(a) \quad \text{and} \quad y = g(a) + tg'(a). \quad (3.2)$$

If $f'(a), g'(a) \neq 0$, we can eliminate the parameter from (3.2)

$$\begin{aligned} \frac{x - f(a)}{f'(a)} &= \frac{y - g(a)}{g'(a)} \\ y - g(a) &= \frac{g'(a)}{f'(a)}(x - f(a)) \end{aligned}$$

and obtain an equation of the tangent line in point-slope form.

$$\begin{aligned} l &= \text{Solve}[x[a] + tx'[a] == cx, t] \\ r &= \text{Solve}[y[a] + ty'[a] == cy, t] \\ \left\{ \left\{ t \rightarrow \frac{cx - x[a]}{x'[a]} \right\} \right\} \\ \left\{ \left\{ t \rightarrow \frac{cy - y[a]}{y'[a]} \right\} \right\} \end{aligned}$$

Example 3.2.8 (The Cycloid). The **cycloid** has parametric equations

$$x = t - \sin t \quad \text{and} \quad y = 1 - \cos t.$$

Graph the cycloid together with the line tangent to the graph of the cycloid at the point $(x(a), y(a))$ for various values of a between -2π and 4π .

Solution After defining x and y , we use $'$ to compute dy/dt and dx/dt . We then compute $dy/dx = (dy/dt)/(dx/dt)$ and d^2y/dx^2 .

$$\begin{aligned} x[t_] &= t - \text{Sin}[t]; \\ y[t_] &= 1 - \text{Cos}[t]; \\ dx &= x'[t] \\ dy &= y'[t] \\ dydx &= dy/dx \\ &= \frac{\text{Sin}[t]}{1 - \text{Cos}[t]} \end{aligned}$$

```
dypdt = Simplify[D[dydx, t]]
```

$$\frac{1}{-1 + \cos[t]}$$

```
secondderiv = Simplify[dypdt/dx]
```

$$-\frac{1}{(-1 + \cos[t])^2}$$

We then use `ParametricPlot` to graph the cycloid for $-2\pi \leq t \leq 4\pi$, naming the resulting graph `p1`.

```
p1 = ParametricPlot[{x[t], y[t]}, {t, -2Pi, 4Pi},
  PlotStyle -> {{Black, Thickness[.01]}}];
```

Next, we use `Table` to define `toplot` to be 40 tangent lines (3.2) using equally spaced values of a between -2π and 4π . We then graph each line `toplot` and name the resulting graph `p2`. Finally, we show `p1` and `p2` together with the `Show` function. The resulting plot is shown to scale because the lengths of the x - and y -axes are equal and we include the option `AspectRatio->1`. In the graphs, notice that on intervals for which dy/dx is defined, dy/dx is a decreasing function and, consequently, $d^2y/dx^2 < 0$. (See Figure 3.15.)

```
toplot = Table[{x[a] + tx'[a], y[a] + ty'[a]}, {a, -2Pi, 4Pi, 6Pi/39}];
p2 = ParametricPlot[Evaluate[toplot], {t, -2, 2}, PlotStyle -> Gray];
Show[p1, p2, AspectRatio -> 1, PlotRange -> {-3Pi, 3Pi}]
```

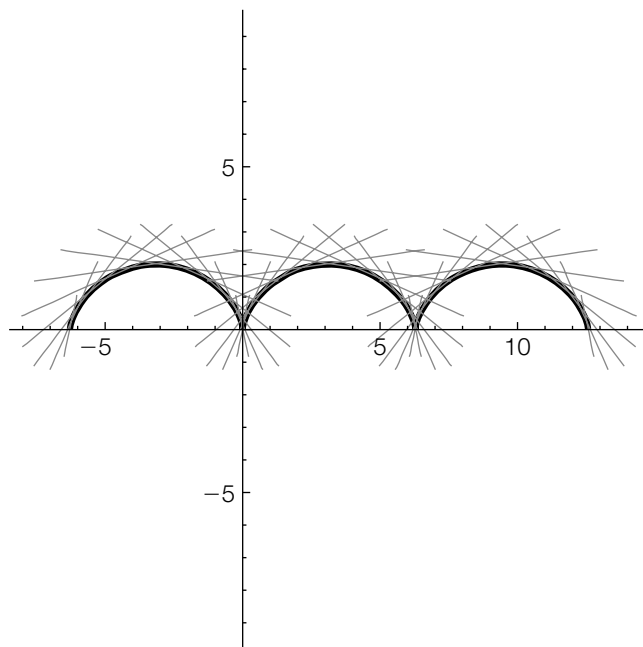


FIGURE 3.15

The cycloid with various tangents

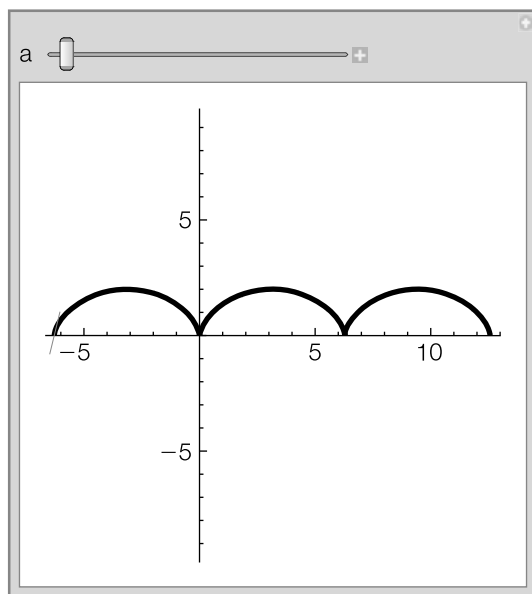


FIGURE 3.16

Using Manipulate you can animate the tangents

With Manipulate, you can animate the the tangents. (See Figure 3.16.)

```
Manipulate[x[t_] = t - Sin[t]; y[t_] = 1 - Cos[t];
  y[t_] = Module[{p1, p2}, p1 = ParametricPlot[{x[t], y[t]}, {t, -2Pi, 4Pi},
    PlotStyle -> {{Black, Thickness[.01]}}];
  p2 = ParametricPlot[{x[a] + tx'[a], y[a] + ty'[a]}, {t, -2, 2},
    PlotStyle -> Gray];
  Show[p1, p2, AspectRatio -> 1, PlotRange -> {{-2Pi, 4Pi}, {-3Pi, 3Pi}}],
  {{a, 1}, -2Pi, 4Pi}]
```

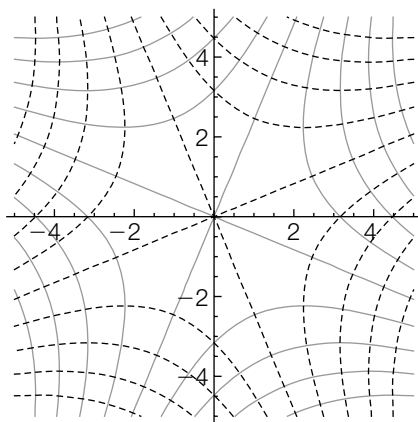
Example 3.2.9 (Orthogonal Curves). Two lines L_1 and L_2 with slopes m_1 and m_2 , respectively, are **orthogonal** if their slopes are negative reciprocals: $m_1 = -1/m_2$.

Extended to curves, we say that the curves C_1 and C_2 are **orthogonal** at a point of intersection if their respective tangent lines to the curves at that point are orthogonal.

Show that the family of curves with equation $x^2 + 2xy - y^2 = C$ is orthogonal to the family of curves with equation $y^2 + 2xy - x^2 = C$.

Solution We begin by defining eq1 and eq2 to be equations $x^2 + 2xy - y^2 = C$ and $y^2 + 2xy - x^2 = C$, respectively. Then, use Dt to differentiate and Solve to find $y' = dy/dx$.

```
eq1 = x^2 + 2xy - y^2 == c;
eq2 = y^2 + 2xy - x^2 == c;
```

**FIGURE 3.17**

$x^2 + 2xy - y^2 = C$ and $y^2 + 2xy - x^2 = C$ for various values of C

Simplify[Solve[Dt[eq1, x], Dt[y, x]]/.Dt[c, x] → 0]

$$\left\{ \left\{ \text{Dt}[y, x] \rightarrow -\frac{x+y}{x-y} \right\} \right\}$$

Simplify[Solve[Dt[eq2, x], Dt[y, x]]/.Dt[c, x] → 0]

$$\left\{ \left\{ \text{Dt}[y, x] \rightarrow \frac{x-y}{x+y} \right\} \right\}$$

Because the derivatives are negative reciprocals, we conclude that the curves are orthogonal. We confirm this graphically by graphing several members of each family with `ContourPlot` and showing the results together. (See Figure 3.17.)

```
cp1 = ContourPlot[x^2 + 2xy - y^2, {x, -5, 5}, {y, -5, 5},
  ContourShading → False];
cp2 = ContourPlot[y^2 + 2xy - x^2, {x, -5, 5}, {y, -5, 5},
  ContourShading → False,
  ContourStyle → Dashing[{0.01}]];
Show[cp1, cp2, Frame → False, Axes → Automatic,
  AxesOrigin → {0, 0}]
```

Theorem 1. (The Mean-Value Theorem for Derivatives) *If $y = f(x)$ is continuous on $[a, b]$ and differentiable on (a, b) , then there is at least one value of c between a and b for which*

$$f'(c) = \frac{f(b) - f(a)}{b - a} \quad \text{or, equivalently,} \quad f(b) - f(a) = f'(c)(b - a). \quad (3.3)$$

Example 3.2.10 Find all number(s) c that satisfy the conclusion of the mean-value theorem for $f(x) = x^2 - 3x$ on the interval $[0, 7/2]$.

Solution By the power rule, $f'(x) = 2x - 3$. The slope of the secant containing $(0, f(0))$ and $(7/2, f(7/2))$ is

$$\frac{f(7/2) - f(0)}{7/2 - 0} = \frac{1}{2}.$$

Solving $2x - 3 = 1/2$ for x gives us $x = 7/4$.

```
f[x_] = x^2 - 3x
```

```
Solve[f'[x]==0, x]
```

```
{{x -> 3/2}}
```

```
Solve[f'[x]==(f[7/2]-f[0])/(7/2-0)]
```

```
{{x -> 7/4}}
```

$x = 7/4$ satisfies the conclusion of the mean-value theorem for $f(x) = x^2 - 3x$ on the interval $[0, 7/2]$, as shown in Figure 3.18.

```
p1 = Plot[f[x], {x, -1, 4}];
p2 = Plot[f[x], {x, 0, 7/2}, PlotStyle -> Thickness[.02]];
p3 = ListPlot[{{0, f[0]}, {7/4, f[7/4]}, {7/2, f[7/2]}},
  PlotStyle -> PointSize[.05]];
p4 = Plot[{f'[7/4](x - 7/4) + f[7/4], (f[7/2] - f[0])/(7/2 - 0)x + f[0]},
  {x, -2, 4}, PlotStyle -> {Dashing[{.01}], Dashing[{.02}]}];
Show[p1, p2, p3, p4, DisplayFunction -> $DisplayFunction,
  AspectRatio -> Automatic, PlotRange -> {-3, 3}]
```

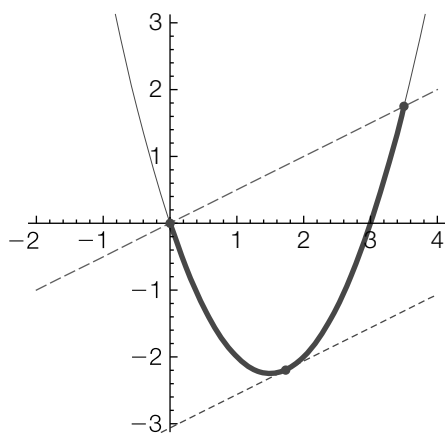


FIGURE 3.18

Graphs of $f(x) = x^2 - 3x$, the secant containing $(0, f(0))$ and $(7/2, f(7/2))$, and the tangent at $(7/4, f(7/4))$

3.2.5 The First Derivative Test and Second Derivative Test

Examples 3.2.11 and 3.2.12 illustrate the following properties of the first and second derivative.

Theorem 2. *Let $y = f(x)$ be continuous on $[a, b]$ and differentiable on (a, b) .*

1. *If $f'(x) = 0$ for all x in (a, b) , then $f(x)$ is constant on $[a, b]$.*
2. *If $f'(x) > 0$ for all x in (a, b) , then $f(x)$ is increasing on $[a, b]$.*
3. *If $f'(x) < 0$ for all x in (a, b) , then $f(x)$ is decreasing on $[a, b]$.*

For the second derivative, we have the following theorem.

Theorem 3. *Let $y = f(x)$ have a second derivative on (a, b) .*

1. *If $f''(x) > 0$ for all x in (a, b) , then the graph of $f(x)$ is concave up on (a, b) .*
2. *If $f''(x) < 0$ for all x in (a, b) , then the graph of $f(x)$ is concave down on (a, b) .*

The **critical points** correspond to those points on the graph of $y = f(x)$ where the tangent line is horizontal or vertical; the number $x = a$ is a **critical number** if $f'(a) = 0$ or $f'(x)$ does not exist if $x = a$. The **inflection points** correspond to those points on the graph of $y = f(x)$ where the graph of $y = f(x)$ is neither concave up nor concave down. Theorems 2 and 3 help establish the first derivative test and second derivative test.

Theorem 4. (First Derivative Test) *Let $x = a$ be a critical number of a function $y = f(x)$ continuous on an open interval I containing $x = a$. If $f(x)$ is differentiable on I , except possibly at $x = a$, $f(a)$ can be classified as follows.*

1. *If $f'(x)$ changes from positive to negative at $x = a$, then $f(a)$ is a **relative maximum**.*
2. *If $f'(x)$ changes from negative to positive at $x = a$, then $f(a)$ is a **relative minimum**.*

Theorem 5. (Second Derivative Test) *Let $x = a$ be a critical number of a function $y = f(x)$ and suppose that $f''(x)$ exists on an open interval containing $x = a$.*

1. *If $f''(a) < 0$, then $f(a)$ is a relative maximum.*
2. *If $f''(a) > 0$, then $f(a)$ is a relative minimum.*

Example 3.2.11 Graph $f(x) = 3x^5 - 5x^3$.

Solution We begin by defining $f(x)$ and then computing and factoring $f'(x)$ and $f''(x)$.

```
f[x_] = 3x^5 - 5x^3;
d1 = Factor[f'[x]]
d2 = Factor[f''[x]]
15(-1 + x)x^2(1 + x)
30x(-1 + 2x^2)
```

By inspection, we see that the critical numbers are $x = 0, 1$, and -1 while $f''(x) = 0$ if $x = 0, 1/\sqrt{2}$, or $-1/\sqrt{2}$. Of course, these values can also be found with `Solve` as done next in `cns` and `ins`, respectively.

```
cns = Solve[d1==0]
ins = Solve[d2==0]
{{x -> -1}, {x -> 0}, {x -> 0}, {x -> 1}}
{{x -> 0}, {x -> -1/sqrt(2)}, {x -> 1/sqrt(2)}}
```

We find the critical and inflection points by using `/.` (**Replace All**) to compute $f(x)$ for each value of x in `cns` and `ins`, respectively. The result means that the critical points are $(0, 0)$, $(1, -2)$ and $(-1, 2)$; the inflection points are $(0, 0)$, $(1/\sqrt{2}, -7\sqrt{2}/8)$, and $(-1/\sqrt{2}, 7\sqrt{2}/8)$. We also see that $f''(0) = 0$, so Theorem 5 cannot be used to classify $f(0)$. On the other hand, $f''(1) = 30 > 0$, and $f''(-1) = -30 < 0$, so by Theorem 5, $f(1) = -2$ is a relative minimum and $f(-1) = 2$ is a relative maximum.

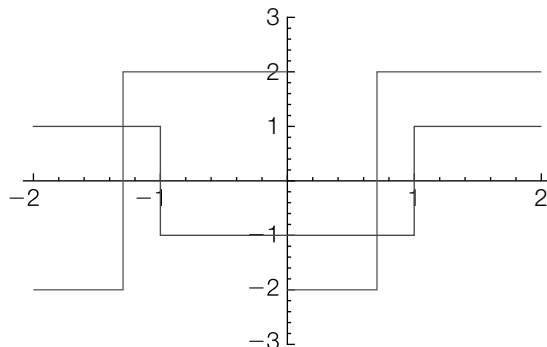
```
cps = {x, f[x]}/.cns
{{-1, 2}, {0, 0}, {0, 0}, {1, -2}}
f''[x]/.cns
{-30, 0, 0, 30}
ips={x, f[x]}/.ins
{{0, 0}, {-1/sqrt(2), 7/(4sqrt(2))}, {1/sqrt(2), -7/(4sqrt(2))}}
```

We can graphically determine the intervals of increase and decrease by noting that if $f'(x) > 0$ ($f'(x) < 0$), $a|f'(x)|/f'(x) = a$ ($a|f'(x)|/f'(x) = -a$). Similarly, the intervals for which the graph is concave up and concave down can be determined by noting that if $f''(x) > 0$ ($f''(x) < 0$), $a|f''(x)|/f''(x) = a$ ($a|f''(x)|/f''(x) = -a$). We use `Plot` to graph $|f'(x)|/f'(x)$ and $2|f''(x)|/f''(x)$ (different values are used so we can differentiate between the two plots) in Figure 3.19.

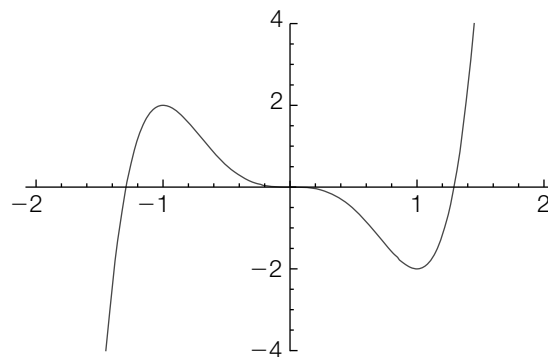
```
Plot[{Abs[d1]/d1, 2Abs[d2]/d2}, {x, -2, 2}, PlotRange -> {-3, 3}]
```

From the graph, we see that $f'(x) > 0$ for x in $(-\infty, -1) \cup (1, \infty)$, $f'(x) < 0$ for x in $(-1, 1)$, $f''(x) > 0$ for x in $(-1/\sqrt{2}, 0) \cup (1/\sqrt{2}, \infty)$, and $f''(x) < 0$ for x in $(-\infty, -1/\sqrt{2}) \cup (0, 1/\sqrt{2})$. Thus, the graph of $f(x)$ is

- increasing and concave down for x in $(-\infty, -1)$,
- decreasing and concave down for x in $(-1, -1/\sqrt{2})$,

**FIGURE 3.19**

Graphs of $|f'(x)|/f'(x)$ and $2|f''(x)|/f''(x)$

**FIGURE 3.20**

$f(x)$ for $-2 \leq x \leq 2$ and $-4 \leq y \leq 4$

- decreasing and concave up for x in $(-1/\sqrt{2}, 0)$,
- decreasing and concave down for x in $(0, 1/\sqrt{2})$,
- decreasing and concave up for x in $(1/\sqrt{2}, 1)$, and
- increasing and concave up for x in $(1, \infty)$.

We also see that $f(0) = 0$ is neither a relative minimum nor maximum. To see all points of interest, our domain must contain -1 and 1 while our range must contain -2 and 2 . We choose to graph $f(x)$ for $-2 \leq x \leq 2$; we choose the range displayed to be $-4 \leq y \leq 4$. (See Figure 3.20.)

Plot[f[x], {x, -2, 2}, PlotRange → {-4, 4}]

Remember to be especially careful when working with functions that involve odd roots.

Example 3.2.12 Graph $f(x) = (x - 2)^{2/3}(x + 1)^{1/3}$.

Solution We begin by defining $f(x)$ and then computing and simplifying $f'(x)$ and $f''(x)$ with $'$ and **Simplify**.

```
Clear[f]
f[x_] = (x - 2)^(2/3)(x + 1)^(1/3);
d1 = Simplify[f'[x]]
d2 = Simplify[f''[x]]

$$\frac{x}{(-2 + x)^{1/3}(1 + x)^{2/3}}$$


$$-\frac{2}{(-2 + x)^{4/3}(1 + x)^{5/3}}$$

```

By inspection, we see that the critical numbers are $x = 0$, 2 , and -1 . We cannot use Theorem 5 to classify $f(2)$ and $f(-1)$ because $f''(x)$ is undefined if $x = 2$ or -1 . On the other hand, $f''(0) < 0$, so $f(0) = 2^{2/3}$ is a relative maximum. By hand, we make a sign chart to see that the graph of $f(x)$ is

- increasing and concave up on $(-\infty, -1)$,
- increasing and concave down on $(-1, 0)$,
- decreasing and concave down on $(0, 2)$, and
- increasing and concave down on $(2, \infty)$.

Hence, $f(-1) = 0$ is neither a relative minimum nor maximum, whereas $f(2) = 0$ is a relative minimum by Theorem 4. To graph $f(x)$, we load the **RealOnly** package and then use **Plot** to graph $f(x)$ for $-2 \leq x \leq 3$ in Figure 3.21.

```
<< Miscellaneous`RealOnly`
f[0]
Plot[f[x], {x, -2, 3}]
 $2^{2/3}$ 
```

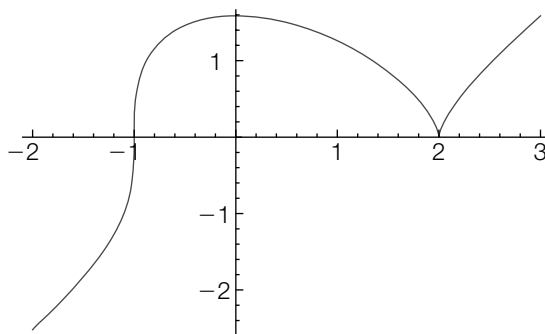


FIGURE 3.21

$f(x)$ for $-2 \leq x \leq 3$

The previous examples illustrate that if $x = a$ is a critical number of $f(x)$ and $f'(x)$ makes a *simple change in sign* from positive to negative at $x = a$, then $(a, f(a))$ is a relative maximum. If $f'(x)$ makes a simple change in sign from negative to positive at $x = a$, then $(a, f(a))$ is a relative minimum. Mathematica is especially useful in investigating interesting functions for which this may not be the case.

Example 3.2.13 Consider

$$f(x) = \begin{cases} x^2 \sin^2\left(\frac{1}{x}\right), & x \neq 0 \\ 0, & x = 0. \end{cases}$$

$x = 0$ is a critical number because $f'(x)$ does not exist if $x = 0$. The point $(0, 0)$ is both a relative and absolute minimum, even though $f'(x)$ does not make a simple change in sign at $x = 0$, as illustrated in Figure 3.22.

```
f[x_] = (xSin[1/x])^2;
f'[x]/Factor
-2Sin[1/x] (Cos[1/x] - xSin[1/x])
p1 = Plot[f[x], {x, -0.1, 0.1}];
p2 = Plot[f'[x], {x, -0.1, 0.1}];
Show[GraphicsRow[{p1, p2}]]
```

Notice that the derivative “oscillates” infinitely many times near $x = 0$, so the first derivative test cannot be used to classify $(0, 0)$.

The functions `Maximize` and `Minimize` can be used to assist with finding extreme values. For a function of a single variable `Maximize[f[x], x]` (`Minimize[f[x], x]`) attempts to find the maximum (minimum) values of $f(x)$;

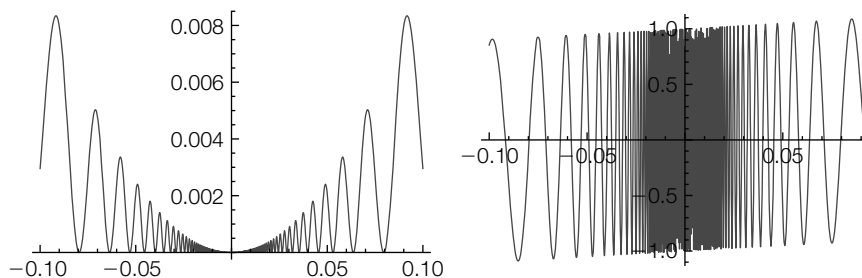


FIGURE 3.22

$f(x) = \left[x \sin\left(\frac{1}{x}\right)\right]^2$ and $f'(x)$ for $-0.1 \leq x \leq 0.1$

`Maximize[f[x], a <= x <= b, x]` (`Minimize[f[x], a <= x <= b, x]`) attempts to find the maximum (minimum) values of $f(x)$ on $[a, b]$.

Example 3.2.14 Consider $f(x) = \frac{1}{10}(-12x + 3x^2 + 2x^3)$. After defining $f(x)$, we plot $f(x)$ and $f'(x)$ together in Figure 3.23.

```
f[x_] = 1/10 (-12x + 3x^2 + 2x^3);
Plot[Tooltip[{f[x], f'[x]}], {x, -4, 4}, PlotRange -> {-4, 4},
     AspectRatio -> Automatic]
```

With `Maximize`, we see that $f(x)$ does not have a maximum on its domain. However, when we restrict the interval to $-3 \leq x \leq 2$, `Maximize` finds the relative maximum at $x = -2$.

```
Maximize[f[x], x]
Maximize::natt: The maximum is not attained at any point satisfying the given constraints.
{∞, {x -> ∞}}

Maximize[{f[x], -3 ≤ x ≤ 2}, x]
{20, {x -> -2}}
```

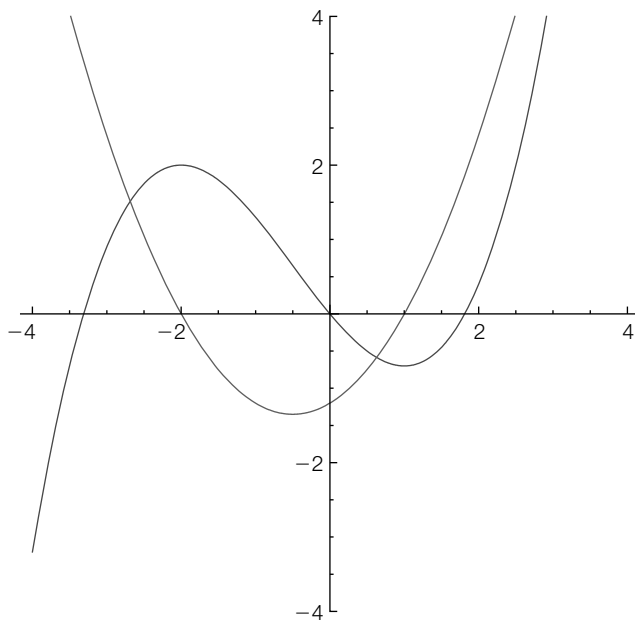


FIGURE 3.23

$f(x)$ has one relative maximum and one relative minimum but no absolute extreme values

Similarly, with **Minimize** we see that the $f(x)$ does not have a minimum value on its domain but find the relative minimum when we restrict the interval to $-3 \leq x \leq 2$.

```
Minimize[f[x], x]
Minimize::natt: The minimum is not attained at any point satisfying the given constraints.
{-∞, {x → -∞}}

Minimize[{f[x], -3 ≤ x ≤ 2}, x]
{-7, {x → 1}}
```

However, with **Solve**, we easily find the two zeros of $f'(x)$ that we see in Figure 3.23.

```
Solve[f'[x]==0,x]
{{x → -2}, {x → 1}}
```

When using **Maximize** or **Minimize** you should verify your results using another method.

Example 3.2.15 The function $f(x) = x/(x^2 + 1)$ is continuous on $(-\infty, \infty)$ and $\lim_{x \rightarrow \pm\infty} f(x) = 0$. Thus, $f(x)$ has an absolute minimum and maximum value on its domain. In this case,

```
Maximize[x / (x^2 + 1), x]
{1/2, {x → 1}}

Minimize[x / (x^2 + 1), x]
{-1/2, {x → -1}}
```

gives us the absolute maximum and minimum values of $f(x)$ and the x -values where they occur. On the other hand, $f(x) = x^4 - x^2$ is continuous on $(-\infty, \infty)$ and $\lim_{x \rightarrow \pm\infty} f(x) = \infty$. Thus, $f(x)$ has an absolute minimum on its domain. Because the derivative of a fourth-degree polynomial is a third-degree polynomial, we know that $f'(x)$ has three zeros, two of which probably correspond to relative minima. Because the graph of $f(x)$ is symmetric with respect to the y -axis, we further suspect that the absolute minimum is obtained twice—at each relative minimum. **Maximize** and **Minimize** give us the following results.

```
Maximize[x^4 - x^2, x]
Maximize::natt: The maximum is not attained at any point satisfying the given constraints.
{∞, {x → ∞}}

Minimize[x^4 - x^2, x]
{-1/4, {x → -1/√2}}
```

A polynomial of degree n has n zeros (counting multiplicity).

Note that the result returned by **Maximize** is correct. Similarly, the result returned by **Minimize** is correct, but a complete answer would indicate that the absolute minimum value occurs at both $x = -1/\sqrt{2}$ and $x = 1/\sqrt{2}$.

Example 3.2.16 The function $f(x) = (x+1)^2/(x-2)$ has a vertical asymptote at $x = 2$. From the derivative,

```
f[x_] = (x + 1)^2/(x - 2);
d1 = Simplify[f'[x]]
cns = Solve[f'[x]==0]
      -5-4x+x^2
      (-2+x)^2
      {{x -> -1}, {x -> 5}}
f[x]/.cns
      {0,12}
```

we find two critical numbers, one of which is a relative maximum and one is a relative minimum. See Figure 3.24.

```
Plot[Tooltip[{f[x], f'[x]}, {x, -6, 10}]
```

On the other hand, **Maximize** and **Minimize** return confusing results because the function is undefined if $x = 2$. The function has relative extreme values but not absolute extreme values.

```
Maximize[f[x], x]
Maximize::natt: The maximum is not attained at any point satisfying the given constraints.
{∞, {x -> 2}}

Minimize[f[x], x]
Minimize::natt: The minimum is not attained at any point satisfying the given constraints.
{-∞, {x -> 2}}
```

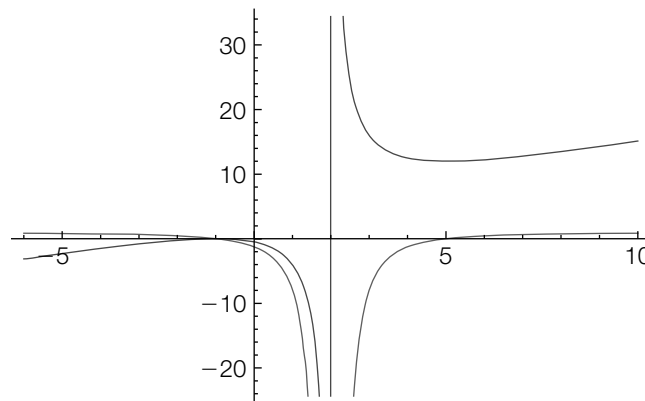


FIGURE 3.24

A function for which a relative minimum has a function value greater than the function value of a relative maximum

For periodic functions, such as sine and cosine, Maximize and Minimize generally do not indicate *all* extreme values.

Maximize[Sin[x], x]

$\{1, \{x \rightarrow \frac{\pi}{2}\}\}$

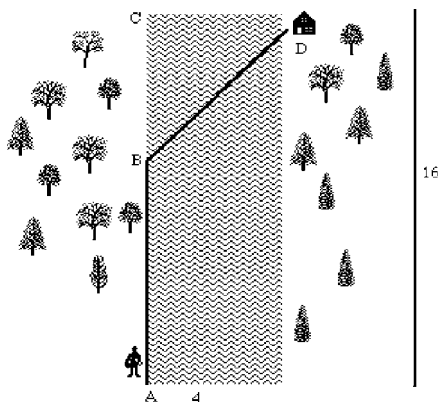
Maximize[Cos[x], x]

$\{1, \{x \rightarrow 0\}\}$

3.2.6 Applied Max/Min Problems

Mathematica can be used to assist in solving maximization/minimization problems encountered in a differential calculus course.

Example 3.2.17 A woman is located on one side of a body of water 4 miles wide. Her position is directly across from a point on the other side of the body of water 16 miles from her house, as shown in the following figure.

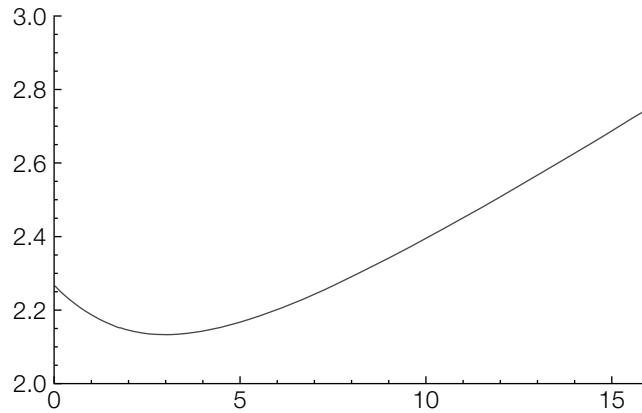


If she can move across land at a rate of 10 miles per hour and move over water at a rate of 6 miles per hour, find the least amount of time for her to reach her house.

Solution From the figure, we see that the woman will travel from A to B by land and then from B to D by water. We wish to find the least time for her to complete the trip.

Let x denote the distance BC , where $0 \leq x \leq 16$. Then, the distance AB is given by $16 - x$ and, by the Pythagorean theorem, the distance BD is given by $\sqrt{x^2 + 4^2}$. Because rate $\times$ time = distance, time = distance / rate. Thus, the time to travel from A to B is $\frac{1}{10}(16 - x)$, the time to travel from B to D is $\frac{1}{6}\sqrt{x^2 + 16}$, and the total time to complete the trip, as a function of x , is

$$\text{time}(x) = \frac{1}{10}(16 - x) + \frac{1}{6}\sqrt{x^2 + 16}, \quad 0 \leq x \leq 16.$$

**FIGURE 3.25**

Plot of $\text{time}(x) = \frac{1}{10}(16 - x) + \frac{1}{6}\sqrt{x^2 + 16}$, $0 \leq x \leq 16$

We must minimize the function **time**. First, we define **time** and then verify that **time** has a minimum by graphing **time** on the interval $[0, 16]$ in Figure 3.25.

Clear[time]

time[x_] = $\frac{16-x}{10} + \frac{1}{6}\sqrt{x^2 + 16}$;

Plot[time[x], {x, 0, 16}, PlotRange → {{0, 16}, {2, 3}}]

Next, we compute the derivative of **time** and find the values of x for which the derivative is 0 with **Solve**. The resulting output is named **critnums** using **ReplaceAll** (**/.**).

Together [time'[x]]

$$\frac{5x - 3\sqrt{16+x^2}}{30\sqrt{16+x^2}}$$

critnums = Solve [time'[x]==0]

{{x → 3}}

At this point, we can calculate the minimum time by calculating **time[3]**.

time[3]

$$\frac{32}{15}$$

Alternatively, we demonstrate how to find the value of **time[x]** for the value(s) listed in **critnums**.

time[x]/.x → 3

$$\frac{32}{15}$$

Regardless, we see that the minimum time to complete the trip is $32/15$ hours.

One of the more interesting applied max/min problems is the *beam problem*. We present two solutions.

Example 3.2.18 (The Beam Problem). Find the exact length of the longest beam that can be carried around a corner from a hallway 2 feet wide to a hallway that is 3 feet wide. (See Figure 3.26.)

Solution We assume that the beam has negligible thickness. Our first approach is algebraic. Using Figure 3.26, which is generated with

Graphics primitives such as Point, Line, and Text are discussed in more detail in Chapter 7.

```
f[x_] = x + 2;
p1 = Plot[f[x], {x, 0, 4}, PlotStyle -> Thickness[.01], PlotRange -> {0, 6};
p2 =
Graphics[Line[{{1, 0}, {1, f[1]}, {4, f[1]}, {4, f[4]}, {4, f[4]},
{0, f[4]},
{0, 0}, {1, 0}}]];
p3 = Graphics[{Text["2", { . 5, .2}], Text["3", {3.8, 4.5}]}];
p4 = Graphics[{Dashing[{0.01, 0.01}], Line[{{0, f[0]}, {1, f[0]}]}];
p5 = Graphics[{Text["θ", { . 5, 2.25}], Text["θ", {1.5, 3.25}]}];
p6 = Graphics[{Text["x", { . 9, 2.35}], Text["y", {2.5, 3.25}]}];
Show[p1, p2, p3, p4, p5, p6, Axes->None]
```

and the Pythagorean theorem, the total length of the beam is

$$L = \sqrt{2^2 + x^2} + \sqrt{y^2 + 3^2}.$$

By similar triangles,

$$\frac{y}{3} = \frac{2}{x} \quad \text{so} \quad y = \frac{6}{x}$$

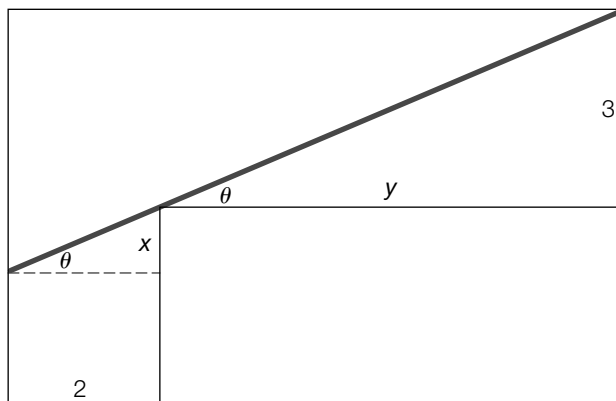


FIGURE 3.26

The length of the beam is found using similar triangles

and the length of the beam, L , becomes

$$L(x) = \sqrt{4 + x^2} + \sqrt{9 + \frac{36}{x^2}}, \quad 0 < x < \infty.$$

Observe that the length of the longest beam is obtained by *minimizing* L . (Why?)

We ignore negative and imaginary values because length must be a nonnegative real number.

```
Clear[l];
l[x_] = Sqrt[2^2 + x^2] + Sqrt[y^2 + 3^2]/.y->6/x
Sqrt[9 + 36/x^2] + Sqrt[4 + x^2]
```

We use two different methods to solve $L'(x) = 0$. Differentiating

```
l'[x]
- 36/(Sqrt[9 + 36/x^2] x^3) + x/Sqrt[4 + x^2]

Solve[-12 Sqrt[4 + x^2] + x^4 Sqrt[4 + x^2]/x^2 == 0, x]
{{x -> -2i}, {x -> 2i}, {x -> -2^(2/3) 3^(1/3)}, {x -> 2^(2/3) 3^(1/3)}}
p1 = x^4 8(9 + 36/x^2) - 1296(4 + x^2) // Expand // Factor
9(4 + x^2)(-12 + x^3)(12 + x^3)
```

and solving $L'(x) = 0$ gives us

```
Solve[p1 == 0, x]
{{x -> -2i}, {x -> 2i}, {x -> -(-3)^(1/3) 2^(2/3)}, {x -> (-3)^(1/3) 2^(2/3)},
{x -> -(-2)^(2/3) 3^(1/3)},
N[2^(2/3) 3^(1/3)]
2.28943
l[2^(2/3) 3^(1/3)]
Sqrt[9 + 32^(2/3) 3^(1/3)] + Sqrt[4 + 22^(1/3) 3^(2/3)]
N[%]
7.02348
```

It follows that the length of the beam is $L(2^{2/3} 3^{1/3}) = \sqrt{9 + 3 \cdot 2^{2/3} \cdot 3^{1/3}} + \sqrt{4 + 2 \cdot 2^{1/3} \cdot 3^{2/3}} = \sqrt{13 + 9 \cdot 2^{2/3} \cdot 3^{1/3} + 6 \cdot 2^{1/3} \cdot 3^{2/3}} \approx 7.02$. (See Figure 3.27).

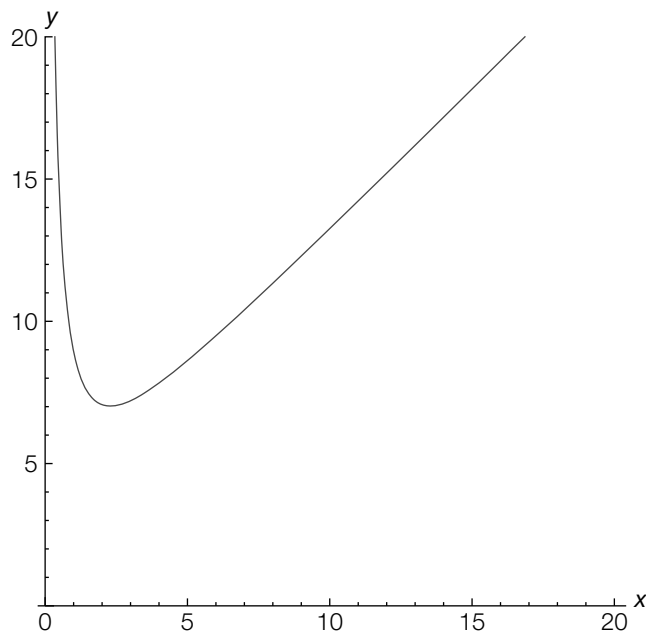
```
Plot[l[x], {x, 0, 20}, PlotRange -> {0, 20}, AspectRatio -> Automatic,
AxesLabel -> {"x", "y"}]
```

Our second approach uses right triangle trigonometry. In terms of θ , the length of the beam is given by

$$L(\theta) = 2 \csc \theta + 3 \sec \theta, \quad 0 < \theta < \pi/2.$$

Differentiating gives us

$$L'(\theta) = -2 \csc \theta \cot \theta + 3 \sec \theta \tan \theta.$$

**FIGURE 3.27**Graph of $L(x)$

To avoid typing the θ symbol, we define L as a function of t .

```
Clear[]
l[t_] = 2Csc[t] + 3Sec[t]
2Csc[t] + 3Sec[t]
```

We now solve $L'(\theta) = 0$. First, multiply through by $\sin \theta$ and then by $\tan \theta$.

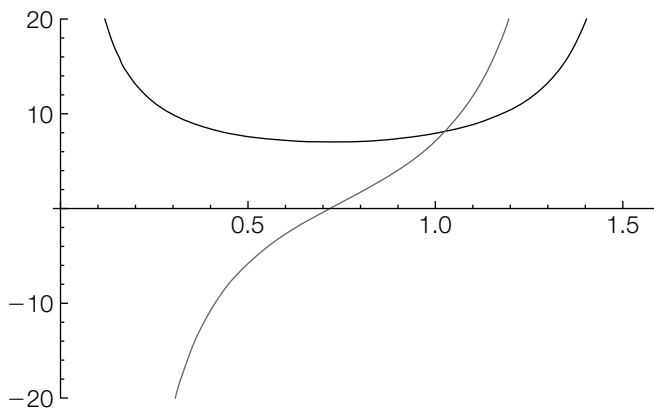
$$3 \sec \theta \tan \theta = 2 \csc \theta \cot \theta$$

$$\tan^2 \theta = \frac{2}{3} \cot \theta$$

$$\tan^3 \theta = \frac{2}{3}$$

$$\tan \theta = \sqrt[3]{\frac{2}{3}}.$$

In this case, observe that we cannot compute θ exactly. However, we do not need to do so. Let $0 < \theta < \pi/2$ be the unique solution of $\tan \theta = \sqrt[3]{2/3}$. See Figure 3.28. Using the identity $\tan^2 \theta + 1 = \sec^2 \theta$, we find that $\sec \theta = \sqrt{1 + \sqrt[3]{4/9}}$. Similarly, because $\cot \theta = \sqrt[3]{3/2}$ and $\cot^2 \theta + 1 = \csc^2 \theta$, $\csc \theta = \sqrt[3]{3/2} \sqrt{1 + \sqrt[3]{4/9}}$. Hence, the length of the beam is

**FIGURE 3.28**Graph of $L(\theta)$ and $L'(\theta)$

When you use Tooltip, scrolling the cursor over the plot will identify the plot for you.

$$L(\theta) = 2\sqrt[3]{\frac{3}{2}}\sqrt{1 + \sqrt[3]{\frac{4}{9}}} + 3\sqrt{1 + \sqrt[3]{\frac{4}{9}}} \approx 7.02.$$

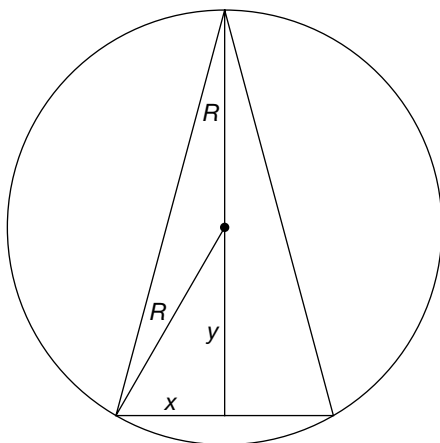
```
Plot[Tooltip[{l[t], l'[t]}, {t, 0, Pi/2}, PlotRange -> {-20, 20},
PlotStyle -> {Black, Gray}]
```

In the next two examples, the constants do not have specific numerical values.

Example 3.2.19 Find the volume of the right circular cone of maximum volume that can be inscribed in a sphere of radius R .

Solution Try to avoid three-dimensional figures unless they are absolutely necessary. For this problem, a cross section of the situation is sufficient. See Figure 3.29, which is created with

```
p1 = ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2Pi}];
p2 = Graphics[
  {Line[{{0, 1}, {Cos[4Pi/3], Sin[4Pi/3]}, {Cos[5Pi/3], Sin[5Pi/3]},
    {0, 1}}, PointSize[.02], Point[{0, 0}],
    Line[{{Cos[4Pi/3], Sin[4Pi/3]}, {0, 0}, {0, 1}},
    Line[{{0, 0}, {0, Sin[4Pi/3]}]}];
p3 = Graphics[{Text["R", {-256, -.28}], Text["R", {-0.04, .5}],
  Text["y", {-0.04, -.5}], Text["x", {-0.2, -.8}]}];
Show[p1, p2, p3, AspectRatio -> Automatic, Ticks -> None, Axes -> None]
```

**FIGURE 3.29**

Cross section of a right circular cone inscribed in a sphere

The volume, V , of a right circular cone with radius r and height h is $V = \frac{1}{3}\pi r^2 h$. Using the notation in Figure 3.29, the volume is given by

$$V = \frac{1}{3}\pi x^2(R + y). \quad (3.4)$$

However, by the Pythagorean theorem, $x^2 + y^2 = R^2$ so $x^2 = R^2 - y^2$ and equation 3.4 becomes

$$V = \frac{1}{3}\pi (R^2 - y^2)(R + y) = \frac{1}{3}\pi (R^3 + R^2y - Ry^2 - y^3), \quad (3.5)$$

s1 = Expand[(r^2 - y^2)(r + y)]

$$r^3 + r^2y - ry^2 - y^3$$

Remember that R is a constant.

where $0 \leq y \leq R$. $V(y)$ is continuous on $[0, R]$, so it will have minimum and maximum values on this interval. Moreover, the minimum and maximum values occur either at the endpoints of the interval or at the critical numbers on the interior of the interval. Differentiating equation (3.5) with respect to y gives us

$$\frac{dV}{dy} = \frac{1}{3}\pi (R^2 - 2Ry - 3y^2) = \frac{1}{3}\pi (R - 3y)(R + y)$$

s2 = D[s1, y]

$$r^2 - 2ry - 3y^2$$

and we see that $dV/dy = 0$ if $y = \frac{1}{3}R$ or $y = -R$.

Factor[s2]

$$(r - 3y)(r + y)$$

Solve[s2==0, y]

$$\{\{y \rightarrow -r\}, \{y \rightarrow \frac{r}{3}\}\}$$

We ignore $y = -R$ because $-R$ is not in the interval $[0, R]$. Note that $V(0) = V(R) = 0$. The maximum volume of the cone is

$$V\left(\frac{1}{3}R\right) = \frac{1}{3}\pi \cdot \frac{32}{27}R^3 = \frac{32}{81}\pi R^3 \approx 1.24R^3.$$

s3=s1/.y->r/3//Together

$$\frac{32r^3}{27}$$

s3*1/3Pi

$$\frac{32\pi r^3}{81}$$

N[%]

$$1.24112r^3$$

Example 3.2.20 (The Stayed-Wire Problem). Two poles D feet apart with heights L_1 feet and L_2 feet are to be stayed by a wire as shown in Figure 3.30. Find the minimum amount of wire required to stay the poles, as illustrated in Figure 3.30, which is generated with

```
p1 = Graphics[Line[{{0, 0}, {0, 4}, {3.5, 0}, {9, 5.5}, {9, 0}, {0, 0}}]];
p2 = Graphics[{Text["L1", { . 2, 2}], Text["L2", {8.8, 2.75}],
  Text["x", {1.75, .2}], Text["x", {1.75, .2}],
```

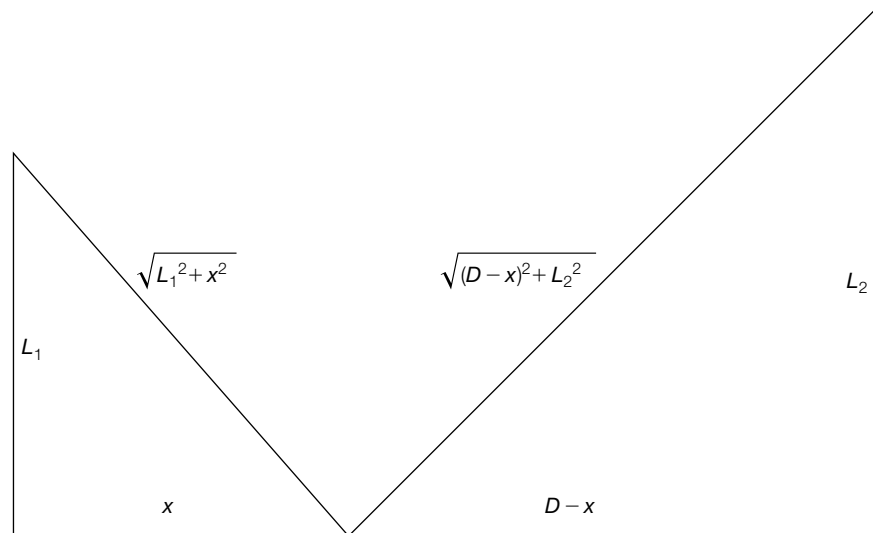


FIGURE 3.30

When the wire is stayed to minimize the length, the result is two similar triangles

Text[" $\sqrt{L_1^2 + x^2}$ ", {1.75, 2.75}], Text[" $\sqrt{(D-x)^2 + L_2^2}$ ", {5.5, 2.75}],
 Text["D - x", {6.5, .2}]]];
 Show[p1, p2]

Solution Using the notation in Figure 3.30, the length of the wire, L , is

$$L(x) = \sqrt{L_1^2 + x^2} + \sqrt{L_2^2 + (D - x)^2}, \quad 0 \leq x \leq D. \quad (3.6)$$

In the special case that $L_1 = L_2$, the length of the wire to stay the beams is minimized when the wire is placed halfway between the two beams, at a distance $D/2$ from each beam. Thus, we assume that the lengths of the beams are different; we assume that $L_1 < L_2$, as illustrated in Figure 3.30. We compute $L'(x)$ and then solve $L'(x) = 0$.

PowerExpand[expr]
 expands out all
 products and powers
 assuming the variables
 are real and positive.
 That is, with
 PowerExpand we
 obtain that $\sqrt{x^2} = x$
 rather than $\sqrt{x^2} = |x|$.

```
Clear[]
l[x_] = Sqrt[x^2 + l1^2] + Sqrt[(d - x)^2 + l2^2]
Sqrt[l2^2 + (d - x)^2] + Sqrt[l1^2 + x^2]
l'[x]//Together
-d*Sqrt[l1^2 + x^2] + x*Sqrt[l1^2 + x^2] + x*Sqrt[d^2 + l2^2 - 2dx + x^2]
Sqrt[l1^2 + x^2]*Sqrt[d^2 + l2^2 - 2dx + x^2]
l[0]//PowerExpand
l1 + Sqrt[d^2 + l2^2]
l[d]//PowerExpand
Sqrt[d^2 + l1^2] + l2
Solve[l'[x]==0, x]
{{x -> d*l1/(l1 + l2)}, {x -> d*l1/(l1 + l2)}}
```

The result indicates that $x = L_1 D / (L_1 + L_2)$ minimizes $L(x)$. (Note that we ignore the other value because $L_1 - L_2 < 0$.) Moreover, the triangles formed by minimizing L are similar triangles.

```
l1/(d*l1/(l1 + l2))//Simplify
l1 + l2
l2/(d - d*l1/(l1 + l2))//Simplify
l1 + l2
```

3.2.7 Antidifferentiation

Antiderivatives

$f(x)$ is an **antiderivative** of $f(x)$ if $F'(x) = f(x)$. The symbol

$$\int f(x) dx$$

means “find all antiderivatives of $f(x)$.” Because all antiderivatives of a given function differ by a constant, we usually find an antiderivative, $f(x)$, of $f(x)$ and then write

$$\int f(x) dx = F(x) + C,$$

where C represents an arbitrary constant. The command

Integrate[f[x],x]

attempts to find an antiderivative, $F(x)$, of $f(x)$. Instead of using **Integrate**, you might prefer to use the  button on the **BasicMathInput** palette to help you evaluate antiderivatives. Mathematica does not include the “+ C ” that we include when writing $\int f(x) dx = f(x) + C$. In the same way as **D** can differentiate many functions, **Integrate** can antidifferentiate many functions. However, antidifferentiation is a fundamentally difficult procedure so it is not difficult to find functions $f(x)$ for which the command **Integrate[f[x],x]** returns unevaluated.

Example 3.2.21 Evaluate each of the following antiderivatives: (a) $\int \frac{1}{x^2} e^{1/x} dx$, (b) $\int x^2 \cos x dx$, (c) $\int x^2 \sqrt{1+x^2} dx$, (d) $\int \frac{x^2 - x + 2}{x^3 - x^2 + x - 1} dx$, and (e) $\int \frac{\sin x}{x} dx$.

Solution Entering

Integrate[1/x^2 Exp[1/x], x]

$-e^{\frac{1}{x}}$

shows us that $\int \frac{1}{x^2} e^{1/x} dx = -e^{1/x} + C$. To use the  button, first click on the button, fill in the blanks, and press Enter.



Notice that Mathematica does not automatically include the arbitrary constant, C . When computing several antiderivatives, you can use **Map** to apply **Integrate** to a list of antiderivatives. However, because **Integrate** is threadable,

Map[Integrate[#, x]&, list]

returns the same result as **Integrate[list, x]**, which we illustrate to compute (b), (c), and (d).

$$\begin{aligned} &\text{Integrate}[(x^2 \cos[x], x^2 \sqrt{1+x^2}, \\ &(x^2 - x + 2)/(x^3 - x^2 + x - 1)), x] \\ &\left\{ 2x \cos[x] + (-2 + x^2) \sin[x], \frac{1}{8} \left(\sqrt{1+x^2} (x+2x^3) - \text{ArcSinh}[x] \right), \right. \\ &\left. -\text{ArcTan}[x] + \text{Log}[-1+x] \right\} \end{aligned}$$

For (e), we see that there is not a “closed form” antiderivative of $\int \frac{\sin x}{x} dx$ and the result is given in terms of a definite integral, the **sine integral function**:

$$\text{Si}(x) = \int_0^x \frac{\sin t}{t} dt.$$

$$\begin{aligned} &\text{Integrate}[\sin[x]/x, x] \\ &\text{SinIntegral}[x] \end{aligned}$$

u-Substitutions

Usually, the first antidifferentiation technique discussed is the method of **u-substitution**. Suppose that $f(x)$ is an antiderivative of $f'(x)$. Given

$$\int f(g(x)) g'(x) dx,$$

we let $u = g(x)$ so that $du = g'(x) dx$. Then,

$$\int f(g(x)) g'(x) dx = \int f(u) du = F(u) + C = F(g(x)) + C,$$

where $F(x)$ is an antiderivative of $f(x)$. After mastering u -substitutions, the **integration by parts formula**,

$$\int u dv = uv - \int v du, \quad (3.7)$$

is introduced.

Example 3.2.22 Evaluate $\int 2^x \sqrt{4^x - 1} dx$.

Solution We use **Integrate** to evaluate the antiderivative. Notice that the result is very complicated.

$$\begin{aligned} &\text{Integrate}[2^x x \sqrt{4^x - 1}, x] \\ &(2^x \\ &\left(2^{1+2x} \text{Log}[2] - \text{Log}[4] - \sqrt{1-4^x} \text{Hypergeometric2F1} \left[\frac{1}{2}, \frac{\text{Log}[2]}{\text{Log}[4]}, \frac{\text{Log}[8]}{\text{Log}[4]}, 4^x \right] \right. \\ &\left. \text{Log}[4] \right) \Big/ \left(\sqrt{-1+4^x} \text{Log}[2] \text{Log}[16] \right) \end{aligned}$$

Proceeding by hand, we let $u = 2^x$. Then, $du = 2^x \ln 2 dx$ or, equivalently, $\frac{1}{\ln 2} du = 2^x dx$

$$D[2^x, x]$$

$$2^x \log[2]$$

so $\int 2^x \sqrt{4^x - 1} dx = \frac{1}{\ln 2} \int \sqrt{u^2 - 1} du$. We now use `Integrate` to evaluate $\int \sqrt{u^2 - 1} du$

$$s1 = \text{Integrate}[\text{Sqrt}[u^2 - 1], u]$$

$$\frac{1}{2} u \sqrt{-1 + u^2} - \frac{1}{2} \text{Log}\left[u + \sqrt{-1 + u^2}\right]$$

and then `/.` `(ReplaceAll)/` to replace u with 2^x .

$$s1 /. u \rightarrow 2^x$$

$$2^{-1+x} \sqrt{-1 + 2^{2x}} - \frac{1}{2} \text{Log}\left[2^x + \sqrt{-1 + 2^{2x}}\right]$$

Clearly, proceeding by hand results in a significantly simpler antiderivative than using `Integrate` directly.

As we did with derivatives, with `DynamicModule`, we create a simple dynamic that lets you compute the derivative and antiderivative of basic functions and plot them on a standard viewing window, $[-5, 5] \times [-5, 5]$. The layout of Figure 3.31 is primarily determined by `Panel`, `Column`, and `Grid`.

```
Panel[DynamicModule[{f = x^2},
  Column[{InputField[Dynamic[f]], Grid[{{"FirstDerivative",
    Panel[Dynamic[D[f, x]//Simplify]],
    {"Antiderivative",
    Panel[Dynamic[Integrate[f, x]//Simplify]]}},
  Dynamic[Plot[Evaluate[Tooltip[{f, D[f, x],
    Integrate[f, x]]], {x, -5, 5}, PlotRange -> {-5, 5},
    AspectRatio -> Automatic]]]], ImageSize -> {300, 300}]
```

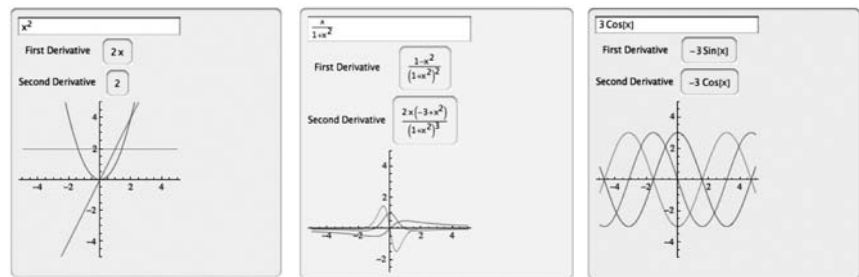


FIGURE 3.31

Seeing the relationship between the derivative and antiderivative of a function and the original function

3.3 INTEGRAL CALCULUS

3.3.1 Area

In integral calculus courses, the definite integral is frequently motivated by investigating the area under the graph of a positive continuous function on a closed interval. Let $y = f(x)$ be a nonnegative continuous function on an interval $[a, b]$ and let n be a positive integer. If we divide $[a, b]$ into n subintervals of equal length and let $[x_{k-1}, x_k]$ denote the k th subinterval, the length of each subinterval is $(b - a)/n$ and $x_k = a + k \frac{b-a}{n}$. The area bounded by the graphs of $y = f(x)$, $x = a$, $x = b$, and the y -axis can be approximated with the sum

$$\sum_{k=1}^n f(x_k^*) \frac{b-a}{n}, \quad (3.8)$$

where $x_k^* \in [x_{k-1}, x_k]$. Typically, we take $x_k^* = x_{k-1} = a + (k-1) \frac{b-a}{n}$ (the left endpoint of the k th subinterval), $x_k^* = x_k = a + k \frac{b-a}{n}$ (the right endpoint of the k th subinterval), or $x_k^* = \frac{1}{2}(x_{k-1} + x_k) = a + \frac{1}{2}(2k-1) \frac{b-a}{n}$ (the midpoint of the k th subinterval). For these choices of x_k^* , (3.8) becomes

$$\frac{b-a}{n} \sum_{k=1}^n f\left(a + (k-1) \frac{b-a}{n}\right) \quad (3.9)$$

$$\frac{b-a}{n} \sum_{k=1}^n f\left(a + k \frac{b-a}{n}\right), \text{ and} \quad (3.10)$$

$$\frac{b-a}{n} \sum_{k=1}^n f\left(a + \frac{1}{2}(2k-1) \frac{b-a}{n}\right), \quad (3.11)$$

respectively. If $y = f(x)$ is increasing on $[a, b]$, (3.9) is an under approximation and (3.10) is an upper approximation: (3.9) corresponds to an approximation of the area using n inscribed rectangles; (3.10) corresponds to an approximation of the area using n circumscribed rectangles. If $y = f(x)$ is decreasing on $[a, b]$, (3.10) is an under approximation and (3.9) is an upper approximation: (3.10) corresponds to an approximation of the area using n inscribed rectangles; (3.9) corresponds to an approximation of the area using n circumscribed rectangles.

In the following example, we define the functions `leftsum[f[x],a,b,n]`, `middlesum[f[x],a,b,n]`, and `rightsum[f[x],a,b,n]` to compute (3.9), (3.11), and (3.10), respectively, and `leftbox[f[x],a,b,n]`, `middlebox[f[x],a,b,n]`, and `rightbox[f[x],a,b,n]` to generate the corresponding graphs. After you have defined these functions, you can use them with functions $y = f(x)$ that you define.

Remark 3.3 To define a function of a single variable, $f(x) = \text{expression in } x$, enter `f[x_]=expression in x`. To generate a basic plot of $y = f(x)$ for $a \leq x \leq b$, enter `Plot[f[x],{x,a,b}]`.

Example 3.3.1 Let $f(x) = 9 - 4x^2$. Approximate the area bounded by the graph of $y = f(x)$, $x = 0$, $x = 3/2$, and the y -axis using (a) 100 inscribed and (b) 100 circumscribed rectangles. (c) What is the exact value of the area?

Solution We begin by defining and graphing $y = f(x)$ in Figure 3.32.

```
f[x_]=9-4x^2;
Plot[f[x],{x,0,3/2}]
```

The first derivative, $f'(x) = -8x$, is negative on the interval, so $f(x)$ is decreasing on $[0, 3/2]$. Thus, an approximation of the area using 100 inscribed rectangles is given by (3.10) whereas an approximation of the area using 100 circumscribed rectangles is given by (3.9). After defining `leftsum`, `rightsum`, and `middlesum`, these values are computed using `leftsum` and `rightsum`. The use of `middlesum` is illustrated as well. Approximations of the sums are obtained with `N`.

```
leftsum[f_, a_, b_, n_] := Module[{ },
  (b - a)/n Sum[f/.x -> a + (k - 1)(b - a)/n, {k, 1, n}];
rightsum[f_, a_, b_, n_] := Module[{ },
  (b - a)/n Sum[f/.x -> a + k(b - a)/n, {k, 1, n}];
middlesum[f_, a_, b_, n_] := Module[{ },
  (b - a)/n Sum[f/.x -> a + 1/2(2k - 1)(b - a)/n, {k, 1, n}];
```

`N[number]` returns a numerical approximation of number.

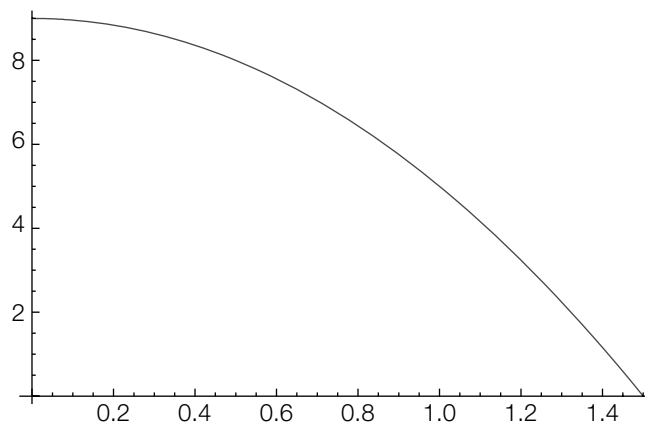


FIGURE 3.32

$f(x)$ for $0 \leq x \leq 3/2$

```

l100 = leftsum[f[x], 0, 3/2, 100]
N[%]
r100 = rightsum[f[x], 0, 3/2, 100]
N[%]
m100 = middlesum[f[x], 0, 3/2, 100]
N[%]

$$\frac{362691}{40000}$$

9.06728

$$\frac{357291}{40000}$$

8.93228

$$\frac{720009}{80000}$$

9.00011

```

Observe that these three values appear to be close to 9. In fact, 9 is the exact value of the area of the region bounded by $y = f(x)$, $x = 0$, $x = 3/2$, and the y -axis. To help us see why this is true, we define `leftbox`, `middlebox`, and `rightbox` and then use these functions to visualize the situation using $n = 4$, 16, and 32 rectangles in Figure 3.33.

It is not important that you understand the syntax of these three functions at this time. Once you have entered the code, you can use them to visualize the process for your own functions, $y = f(x)$.

```

leftbox[f_, a_, b_, n_, opts_>>] := Module[{z, p1, recs, ls},
  z[k_] = a + (b - a)k/n;
  p1 = Plot[f, {x, a, b}, PlotRange -> All,
    PlotStyle -> {{Thickness[.01], GrayLevel[.3]}},
    DisplayFunction -> Identity];
  recs = Table[Rectangle[{z[k - 1], 0}, {z[k], f/.x -> z[k - 1]}], {k, 1, n}];
  ls =
    Table[Line[{{z[k - 1], 0}, {z[k - 1], f/.x -> z[k - 1]}},
      {z[k], f/.x -> z[k - 1]}},
      {z[k], 0}], {k, 1, n}];
  Show[Graphics[{GrayLevel[.8], recs}], Graphics[ls], p1, opts,
    Axes -> Automatic, AspectRatio -> 1,
    DisplayFunction -> $DisplayFunction]]

rightbox[f_, a_, b_, n_, opts_>>] := Module[{z, p1, recs, ls},
  z[k_] = a + (b - a)k/n;
  p1 = Plot[f, {x, a, b}, PlotRange -> All,
    PlotStyle -> {{Thickness[.01], GrayLevel[.3]}},
    DisplayFunction -> Identity];
  recs = Table[Rectangle[{z[k - 1], 0}, {z[k], f/.x -> z[k]}],
    {k, 1, n}];
  ls =
    Table[Line[{{z[k - 1], 0}, {z[k - 1], f/.x -> z[k]}, {z[k], f/.x -> z[k]}},
      {z[k], 0}], {k, 1, n}];
  Show[Graphics[{GrayLevel[.8], recs}], Graphics[ls], p1, opts,
    Axes -> Automatic, AspectRatio -> 1,
    DisplayFunction -> $DisplayFunction]]

```

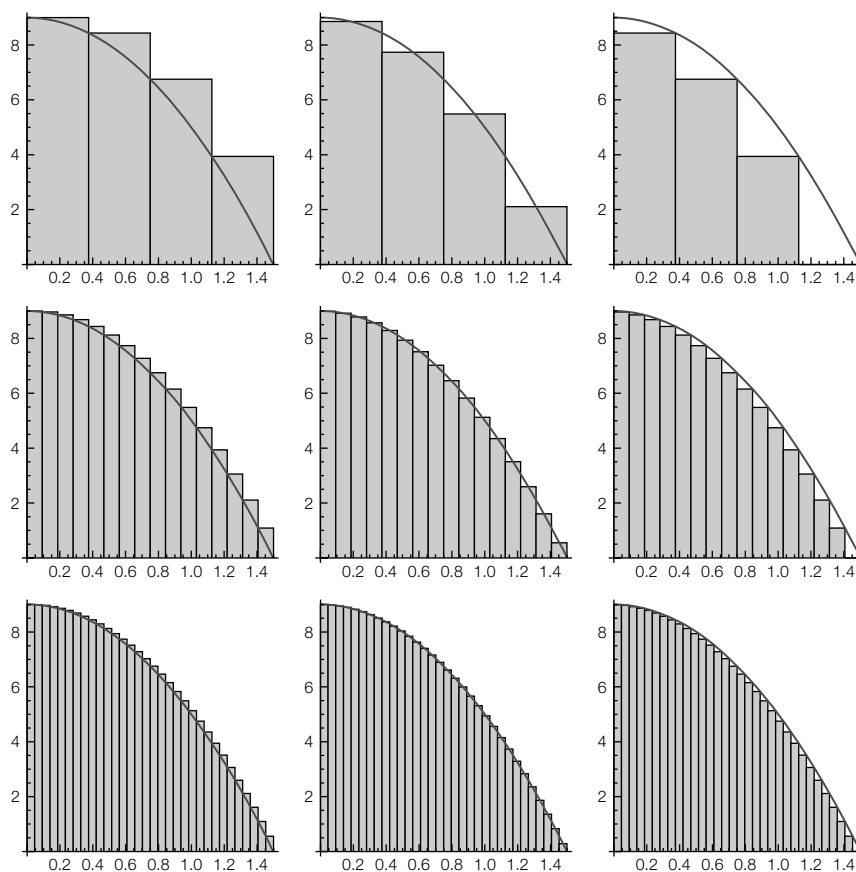


FIGURE 3.33

$f(x)$ with 4, 16, and 32 rectangles

```

middlebox[f_, a_, b_, n_, opts____] := Module[{z, p1, recs, ls},
  z[k_] = a + (b - a)k/n;
  p1 = Plot[f, {x, a, b}, PlotRange -> All,
    PlotStyle -> {{Thickness[.01], GrayLevel[.3]}},
    DisplayFunction -> Identity];
  recs = Table[Rectangle[{z[k - 1], 0}, {z[k], f/.x -> 1/2(z[k - 1] + z[k])}],
    {k, 1, n}];
  ls = Table[Line[{{z[k - 1], 0}, {z[k - 1], f/.x -> 1/2(z[k - 1] + z[k])}},
    {z[k], f/.x -> 1/2(z[k - 1] + z[k])}, {z[k], 0}], {k, 1, n}];
  Show[Graphics[{GrayLevel[.8], recs}], Graphics[ls], p1, opts,
    Axes -> Automatic, AspectRatio -> 1,
    DisplayFunction -> $DisplayFunction]

```

```

somegraphs = {{leftbox [f[x], 0,  $\frac{3}{2}$ , 4, DisplayFunction → Identity] ,
middlebox [f[x], 0,  $\frac{3}{2}$ , 4, DisplayFunction → Identity] ,
rightbox [f[x], 0,  $\frac{3}{2}$ , 4, DisplayFunction → Identity] },
{leftbox [f[x], 0,  $\frac{3}{2}$ , 16, DisplayFunction → Identity] ,
middlebox [f[x], 0,  $\frac{3}{2}$ , 16, DisplayFunction → Identity] ,
rightbox [f[x], 0,  $\frac{3}{2}$ , 16, DisplayFunction → Identity] },
{leftbox [f[x], 0,  $\frac{3}{2}$ , 32, DisplayFunction → Identity] ,
middlebox [f[x], 0,  $\frac{3}{2}$ , 32, DisplayFunction → Identity] ,
rightbox [f[x], 0,  $\frac{3}{2}$ , 32, DisplayFunction → Identity] }};
Show[GraphicsGrid[somegraphs]]

```

Notice that as n increases, the under approximations increase while the upper approximations decrease.

These graphs help convince us that the limit of the sum as $n \rightarrow \infty$ of the areas of the inscribed and circumscribed rectangles is the same. We compute the exact value of Eq. (3.9) with `leftsum`, evaluate and simplify the sum with `Simplify`, and compute the limit as $n \rightarrow \infty$ with `Limit`. We see that the limit is 9.

```

ls = leftsum[f[x], 0, 3/2, n]
ls2 = Simplify[ls]
Limit[ls2, n -> Infinity]

$$\frac{9(1+n)(-1+4n)}{4n^2}$$


$$\frac{9(1+n)(-1+4n)}{4n^2}$$

9

```

Similar calculations are carried out for (3.10) and again we see that the limit is 9. We conclude that the exact value of the area is 9.

```

rs = rightsum[f[x], 0, 3/2, n]
rs2 = Simplify[rs]
Limit[rs2, n -> Infinity]

$$\frac{9(-1+n)(1+4n)}{4n^2}$$


$$\frac{9(-1+n)(1+4n)}{4n^2}$$

9

```

For illustrative purposes, we confirm this result with `middlesum`.

```

ms = middlesum[f[x], 0, 3/2, n]
ms2 = Simplify[ms]
Limit[ms2, n -> Infinity]

$$\frac{9(1+8n^2)}{8n^2}$$


$$9 + \frac{9}{8n^2}$$

9

```

As illustrated previously, with `Manipulate`, you can experiment with different functions and different n values. First, we define a set of “typical” functions.

```

quad[x_] = 100 - x^2;
cubic[x_] = 4/9 x^3 - 49/9 x^2 + 100;
rational[x_] = 100/(x^2 + 1);
root[x_] = Sqrt[10 - x];
sin[x_] = 75Sin[Pi x/5];

```

Next, we use `Manipulate` to create an object that allows us to experiment with how “typical” functions react to changes in n using left, middle, and right-hand endpoint approximations for computations of Riemann sums. In the resulting `Manipulate` object, $n = 4$ rectangles is the default; you can choose n -values from 0 to 100. The value of the corresponding Riemann sum is shown below the graphic. See Figure 3.34.

How does the `Manipulate` object change if you remove `Transpose` from the command?

```

Manipulate[Show[GraphicsGrid[{{leftbox[f[x], 0, 10, n],
Graphics[{Inset[leftsum[f[x], 0, 10, n]/N, {0, 0}]]}},
{middlebox[f[x], 0, 10, n],
Graphics[{Inset[middlesum[f[x], 0, 10, n]/N, {0, 0}]]}},

```

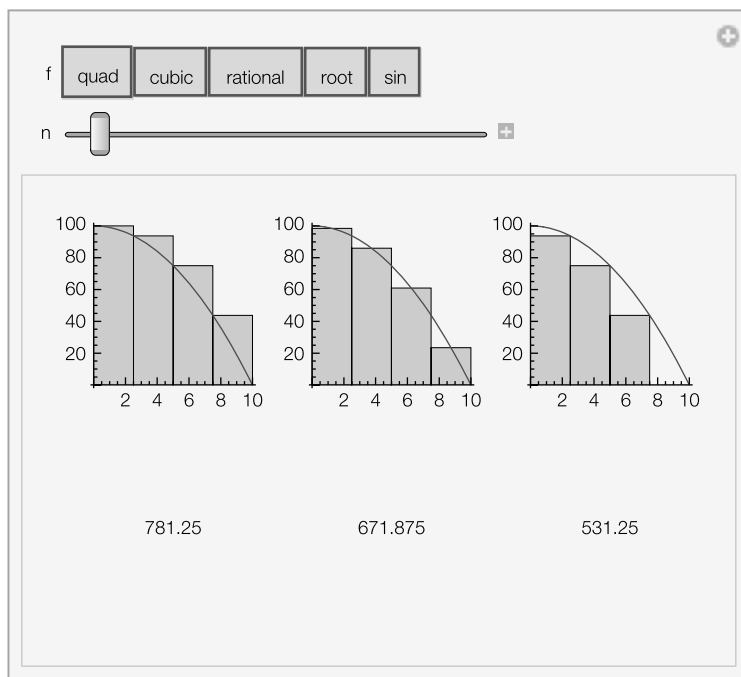


FIGURE 3.34

With `Manipulate`, we can investigate Riemann sum approximations and their graphical representations for various functions

```
{rightbox[f[x], 0, 10, n],
Graphics[{Inset[rightsum[f[x], 0, 10, n] / N, {0, 0}]]}]//
Transpose]], {{f, quad}, {quad, cubic, rational, root, sin}},
{{n, 4}, 0, 100, 1}]
```

3.3.2 The Definite Integral

In integral calculus courses, we formally learn that the **definite integral** of the function $y = f(x)$ from $x = a$ to $x = b$ is

$$\int_a^b f(x) dx = \lim_{|P| \rightarrow 0} \sum_{k=1}^n f(x_k^*) \Delta x_k, \quad (3.12)$$

provided that the limit exists. In equation (3.12), $P = \{a = x_0 < x_1 < x_2 < \dots < x_n = b\}$ is a partition of $[a, b]$, $|P|$ is the **norm** of P ,

$$|P| = \max\{x_k - x_{k-1} | k = 1, 2, \dots, n\},$$

$\Delta x_k = x_k - x_{k-1}$, and $x_k^* \in [x_{k-1}, x_k]$.

The Fundamental Theorem of Calculus provides the fundamental relationship between differentiation and integration.

Theorem 6. (The Fundamental Theorem of Calculus). Suppose that $y = f(x)$ is continuous on $[a, b]$.

1. If $f(x) = \int_a^x f(t) dt$, then F is an antiderivative of f : $F'(x) = f(x)$.
2. If G is any antiderivative of f , then $\int_a^b f(x) dx = G(b) - G(a)$.

Mathematica's Integrate command can compute many definite integrals. The command

```
Integrate[f[x],{x,a,b}]
```

attempts to compute $\int_a^b f(x) dx$. Because integration is a fundamentally difficult procedure, it is easy to create integrals for which the exact value cannot be found explicitly. In those cases, use N to obtain an approximation of its value or obtain a numerical approximation of the integral directly with

```
NIntegrate[f[x],{x,a,b}]
```

In the same way as you use the  button to compute antiderivatives, you can use the  button to compute definite integrals. If the result returned is unevaluated, use N to obtain a numerical approximation of the value of the integral or use NIntegrate.

Example 3.3.2 Evaluate (a) $\int_1^4 (x^2 + 1) / \sqrt{x} \, dx$; (b) $\int_0^{\sqrt{\pi/2}} x \cos x^2 \, dx$; (c) $\int_0^\pi e^{2x} \sin^2 2x \, dx$; (d) $\int_0^1 \frac{2}{\sqrt{\pi}} e^{-x^2} \, dx$; and (e) $\int_{-1}^0 \sqrt[3]{u} \, du$.

Solution We evaluate (a)–(c) directly with `Integrate`.

```
Integrate[(x^2 + 1)/Sqrt[x], {x, 1, 4}]
```

```
 $\frac{72}{5}$ 
```

```
Integrate[x Cos[x^2], {x, 0, Sqrt[Pi/2]]]
```

```
 $\frac{1}{2}$ 
```

```
Integrate[Exp[2x] Sin[2x]^2, {x, 0, Pi}]
```

```
 $\frac{1}{5} (-1 + e^{2\pi})$ 
```

For (d), the result returned is in terms of the **error function**, `Erf[x]`, which is defined by the integral

$$\text{Erf}[x] = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \, dt.$$

```
Integrate[2/Sqrt[Pi] Exp[-x^2], {x, 0, 1}]
```

```
Erf[1]
```

We use `N` to obtain an approximation of the value of the definite integral.

```
Integrate[2/Sqrt[Pi] Exp[-x^2], {x, 0, 1}]/N
```

```
0.842701
```

(e) Recall that Mathematica does not return a real number when we compute odd roots of negative numbers, so the following result would be surprising to many students in an introductory calculus course because it is complex.

```
Integrate[u^(1/3), {u, -1, 0}]
```

```
 $\frac{3}{4} (-1)^{1/3}$ 
```

Therefore, we load the **RealOnly** package contained in the **Miscellaneous** directory so that Mathematica returns the real-valued third root of u .

```
<< Miscellaneous`RealOnly`
```

```
Integrate[u^(1/3), {u, -1, 0}]
```

```
 $-\frac{3}{4}$ 
```

Improper integrals are computed using `Integrate` in the same way as other definite integrals.

See Chapter 2,
Example 2.1.3.

Example 3.3.3 Evaluate (a) $\int_0^1 \frac{\ln x}{\sqrt{x}} dx$; (b) $\int_{-\infty}^{\infty} \frac{2}{\sqrt{\pi}} e^{-x^2} dx$; (c) $\int_1^{\infty} \frac{1}{x\sqrt{x^2-1}} dx$; (d) $\int_0^{\infty} \frac{1}{x^2+x^4} dx$; (e) $\int_2^4 \frac{1}{\sqrt[3]{(x-3)^2}} dx$; and (f) $\int_{-\infty}^{\infty} \frac{1}{x^2+x-6} dx$.

Solution (a) This is an improper integral because the integrand is discontinuous on the interval $[0, 1]$ but we see that the improper integral converges to -4 .

Integrate[Log[x]/Sqrt[x], {x, 0, 1}]

-4

(b) This is an improper integral because the interval of integration is infinite but we see that the improper integral converges to 2.

Integrate[2/Sqrt[Pi]Exp[-x^2], {x, -Infinity, Infinity}]

2

(c) This is an improper integral because the integrand is discontinuous on the interval of integration and because the interval of integration is infinite but we see that the improper integral converges to $\pi/2$.

Integrate[1/(x, Sqrt[x^2-1]), {x, 1, Infinity}]

$\frac{\pi}{2}$

(d) As with (c), this is an improper integral because the integrand is discontinuous on the interval of integration and because the interval of integration is infinite but we see that the improper integral diverges to ∞ .

Integrate[1/(x^2+x^4), {x, 0, Infinity}]

Integrate::idiv: Integral of $\frac{1}{x^2+x^4}$ does not converge on $\{0, \infty\}$. >>

Integrate::idiv: Integral of $\frac{1}{x^2+x^4}$ does not converge on $\{0, \infty\}$. >>

Integrate::idiv: Integral of $\frac{1}{x^2+x^4}$ does not converge on $\{0, \infty\}$. >>

General::stop:

Further output of Integrate::idiv will be suppressed during this calculation. >>

$\int_0^{\infty} \frac{1}{x^2+x^4} dx$

(e) Recall that Mathematica does not return a real number when we compute odd roots of negative numbers, so the following result would be surprising to many students in an introductory calculus course because it contains imaginary numbers.

Integrate[1/(x-3)^(2/3), {x, 2, 4}]

$3-3(-1)^{1/3}$

Therefore, we load the **RealOnly** package contained in the **Miscellaneous** directory so that Mathematica returns the real-valued third root of $x-3$.

You do not need to reload the **RealOnly** package if you have already loaded it during your *current* Mathematica session.

<< Miscellaneous`RealOnly`

Integrate[1/(x-3)^(2/3),{x,2,4}]

6

(f) In this case, Mathematica warns us that the improper integral diverges.

```
Integrate[1/(x^2+x^4), {x, 0, Infinity}]

Integrate::idiv: Integral of  $\frac{1}{x^2+x^4}$  does not converge on {0, ∞}. >>

Integrate::idiv: Integral of  $\frac{1}{x^2+x^4}$  does not converge on {0, ∞}. >>

Integrate::idiv: Integral of  $\frac{1}{x^2+x^4}$  does not converge on {0, ∞}. >>

General::stop:
Further output of Integrate::idiv will be suppressed during this calculation. >>


$$\int_0^{\infty} \frac{1}{x^2+x^4} dx$$

```

To help us understand why the improper integral diverges, we note that

$$\frac{1}{x^2+x-6} = \frac{1}{5} \left(\frac{1}{x-2} - \frac{1}{x+3} \right) \text{ and}$$

$$\int \frac{1}{x^2+x-6} dx = \int \frac{1}{5} \left(\frac{1}{x-2} - \frac{1}{x+3} \right) dx = \frac{1}{5} \ln \left(\frac{x-2}{x+3} \right) + C.$$

Integrate[1/(x^2+x-6),x] $\frac{1}{5} \text{Log}[-2+x] - \frac{1}{5} \text{Log}[3+x]$

Hence, the integral is improper because the interval of integration is infinite and because the integrand is discontinuous on the interval of integration so

```
s1 = Integrate[1/(x^2+x-6), {x, -Infinity, Infinity}]

Integrate::idiv: Integral of  $\frac{1}{-6+x+x^2}$  does not converge on {-∞, ∞}. >>


$$\int_{-\infty}^{\infty} \frac{1}{-6+x+x^2} dx$$


Integrate[1/(x^2+x-6), x]


$$\frac{1}{5} \text{Log}[-2+x] - \frac{1}{5} \text{Log}[3+x]$$


Integrate[1/(x^2+x-6), {x, -Infinity, -4}]


$$\frac{\text{Log}[6]}{5}$$


Integrate[1/(x^2+x-6), {x, -4, -3}]

Integrate::idiv: Integral of  $\frac{1}{-6+x+x^2}$  does not converge on {-4, -3}. >>


$$\int_{-4}^{-3} \frac{1}{-6+x+x^2} dx$$

```

$$\begin{aligned}
 \int_{-\infty}^{\infty} \frac{1}{x^2 + x - 6} dx &= \int_{-\infty}^{-4} \frac{1}{x^2 + x - 6} dx + \int_{-4}^{-3} \frac{1}{x^2 + x - 6} dx \\
 &\quad + \int_{-3}^0 \frac{1}{x^2 + x - 6} dx + \int_0^2 \frac{1}{x^2 + x - 6} dx \\
 &\quad + \int_2^3 \frac{1}{x^2 + x - 6} dx + \int_3^{\infty} \frac{1}{x^2 + x - 6} dx.
 \end{aligned} \tag{3.13}$$

Evaluating each of these integrals,

```

Integrate[1 / (x^2 + x - 6), {x, -3, 0}]

Integrate::idiv: Integral of 1/(-6 + x + x^2) does not converge on {-3, 0}. >>

∫-30 1/(-6 + x + x2) dx

Integrate[1 / (x^2 + x - 6), {x, 0, 2}]

Integrate::idiv: Integral of 1/(-6 + x + x^2) does not converge on {0, 2}. >>

∫02 1/(-6 + x + x2) dx

Integrate[1 / (x^2 + x - 6), {x, 2, 3}]

Integrate::idiv: Integral of 1/(-6 + x + x^2) does not converge on {2, 3}. >>

∫23 1/(-6 + x + x2) dx

Integrate[1 / (x^2 + x - 6), {x, 3, Infinity}]

Log[6]
5

```

we conclude that the improper integral diverges because at least one of the improper integrals in (3.13) diverges.

In many cases, Mathematica can help illustrate the steps carried out when computing integrals using standard methods of integration such as u -substitutions and integration by parts.

Example 3.3.4 Evaluate (a) $\int_e^{e^3} \frac{1}{x\sqrt{\ln x}} dx$ and (b) $\int_0^{\pi/4} x \sin 2x dx$.

Solution (a) We let $u = \ln x$. Then, $du = 1/x dx$ so $\int_e^{e^3} \frac{1}{x\sqrt{\ln x}} dx = \int_1^3 \frac{1}{\sqrt{u}} du = \int_1^3 u^{-1/2} du$, which we evaluate with Integrate.

```

Integrate[1/Sqrt[u], {u, 1, 3}]
2(-1 + Sqrt[3])

```

The new lower limit of integration is 1 because if $x = e$, $u = \ln e = 1$. The new upper limit of integration is 3 because if $x = e^3$, $u = \ln e^3 = 3$.

To evaluate (b), we let $u = x \Rightarrow du = dx$ and $dv = \sin 2x \, dx \Rightarrow v = -\frac{1}{2} \cos 2x$.

```
u = x;
dv = Sin[2x];
du = D[x, x]
v = Integrate[Sin[2x], x]
1
- 1/2 Cos[2x]
```

The results mean that

$$\begin{aligned} \int_0^{\pi/4} x \sin 2x \, dx &= -\frac{1}{2} x \cos 2x \Big|_0^{\pi/4} + \frac{1}{2} \int_0^{\pi/4} \cos 2x \, dx \\ &= 0 + \frac{1}{2} \int_0^{\pi/4} \cos 2x \, dx. \end{aligned}$$

The resulting indefinite integral is evaluated with `Integrate`.

```
Integrate[x Sin[2x], x]
- 1/2 x Cos[2x] + 1/4 Sin[2x]
```

Alternatively, we can illustrate the integration by parts calculation.

```
u = x;
dv = Sin[2x];
du = D[x, x]
v = Integrate[Sin[2x], x]
1
- 1/2 Cos[2x]
```

```
uv - Integrate[v du, x]
- 1/2 x Cos[2x] + 1/4 Sin[2x]
```

We use `Integrate` to evaluate the definite integral.

```
Integrate[x Sin[2x], {x, 0, Pi/4}]
1/4
```

3.3.3 Approximating Definite Integrals

Because integration is a fundamentally difficult procedure, Mathematica is unable to compute a “closed form” of the value of many definite integrals. In these cases, numerical integration can be used to obtain an approximation of the definite integral using `N` together with `Integrate` or `NIntegrate`:

```
NIntegrate[f[x], {x, a, b}]
```

attempts to approximate $\int_a^b f(x) \, dx$.

Example 3.3.5 Evaluate $\int_0^{\sqrt[3]{\pi}} e^{-x^2} \cos x^3 dx$.

Solution In this case, Mathematica is unable to evaluate the integral with `Integrate`.

We use the  button to complete the `Integrate` command.

```
i1 = Integrate[Exp[-x^2]Cos[x^3], {x, 0, Pi^(1/3)}]
```

An approximation is obtained with `N`.

```
N[i1]
```

0.701566

Instead of using `Integrate` followed by `N`, you can use `NIntegrate` to numerically evaluate the integral.

```
NIntegrate[Exp[-x^2]Cos[x^3], {x, 0, Pi^(1/3)}]
```

0.701566

returns the same result as that obtained using `Integrate` followed by `N`.

In some cases, you may wish to investigate particular numerical methods that can be used to approximate integrals. To implement numerical methods such as Simpson's rule or the trapezoidal rule, redefine the function `leftsum` (middlesum or `rightsum`) discussed previously to perform the calculation for the desired method.

3.3.4 Area

Suppose that $y = f(x)$ and $y = g(x)$ are continuous on $[a, b]$ and that $f(x) \geq g(x)$ for $a \leq x \leq b$. The **area** of the region bounded by the graphs of $y = f(x)$, $y = g(x)$, $x = a$, and $x = b$ is

$$A = \int_a^b [f(x) - g(x)] dx. \quad (3.14)$$

Example 3.3.6 Find the area between the graphs of $y = \sin x$ and $y = \cos x$ on the interval $[0, 2\pi]$.

Solution We graph $y = \sin x$ and $y = \cos x$ on the interval $[0, 2\pi]$ in Figure 3.35 with `Plot`. The graph of $y = \cos x$ is dashed. Observe that including the option `Filling->{1->{2}}` fills the region *between* the two plots.

We display a portion of Mathematica's error message because it indicates that Mathematica cannot find all solutions of the equation. In this case, $\sin x = \cos x$ has infinitely many solutions.

```
Plot[{Sin[x], Cos[x]}, {x, 0, 2 Pi}, PlotStyle -> {Black, Dashing[{0.01]}],  
Filling -> {1 -> {2}}, AspectRatio -> Automatic]
```

To find the upper and lower limits of integration, we must solve the equation $\sin x = \cos x$ for x .

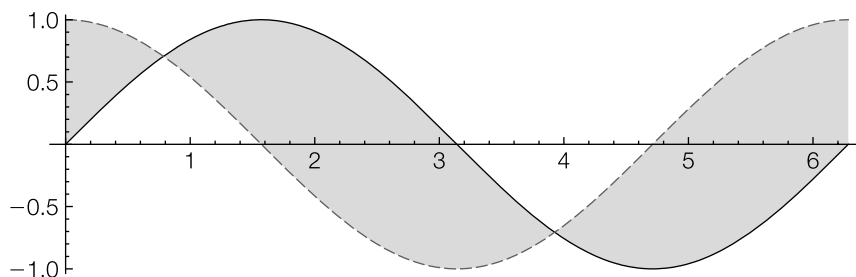


FIGURE 3.35

$y = \sin x$ and $y = \cos x$ on the interval $[0, 2\pi]$.

```
Solve[Sin[x] == Cos[x], x]
Solve::ifun:
Inverse functions are being used by Solve, so some solutions may not
be found; use Reduce for complete solution information. >>
{{x -> -3 π/4}, {x -> π/4}}
```

For us the solutions of interest are valid for $0 \leq x \leq 2\pi$, which are $x = \pi/4$ and $x = 5\pi/4$. We check that these are valid solutions of $\sin x = \cos x$ with `==`; in each case the returned result is `True`.

```
Sin[π/4]==Cos[π/4]
```

```
Sin[5π/4]==Cos[5π/4]
```

```
True
```

```
True
```

Hence, the area of the region between the graphs is given by

$$A = \int_0^{\pi/4} [\cos x - \sin x] dx + \int_{\pi/4}^{5\pi/4} [\sin x - \cos x] dx + \int_{5\pi/4}^{2\pi} [\cos x - \sin x] dx. \quad (3.15)$$

Notice that if we take advantage of symmetry we can simplify (3.15) to

$$A = 2 \int_{\pi/4}^{5\pi/4} [\sin x - \cos x] dx. \quad (3.16)$$

We evaluate (3.16) with `Integrate` to see that the area of the region between the two graphs is $4\sqrt{2}$.

$$\int_0^{\pi/4} (\cos[x] - \sin[x]) dx + \int_{\pi/4}^{5\pi/4} (\sin[x] - \cos[x]) dx + \int_{5\pi/4}^{2\pi} (\cos[x] - \sin[x]) dx$$

$4\sqrt{2}$

In cases in which we cannot calculate the points of intersection of two graphs exactly, we can frequently use FindRoot to approximate the points of intersection.

Example 3.3.7 Let

$$p(x) = \frac{3}{10}x^5 - 3x^4 + 11x^3 - 18x^2 + 12x + 1$$

and

$$q(x) = -4x^3 + 28x^2 - 56x + 32.$$

Approximate the area of the region bounded by the graphs of $y = p(x)$ and $y = q(x)$.

Solution After defining p and q , we graph them on the interval $[-1, 5]$ in Figure 3.36 to obtain an initial guess of the intersection points of the two graphs.

When you use Tooltip, you can slide your cursor over a plot and the function being graphed is displayed.

```
Clear[p,q]
p[x_] =  $\frac{3x^5}{10} - 3x^4 + 11x^3 - 18x^2 + 12x + 1$ ;
q[x_] =  $-4x^3 + 28x^2 - 56x + 32$ ;
Plot[Tooltip[{p[x], q[x]}], {x, -1, 5}, PlotStyle -> {Black, Gray}]
```

The x -coordinates of the three intersection points are the solutions of the equation $p(x) = q(x)$. Although Mathematica can solve this equation exactly, approximate solutions are more useful for the problem and obtained with NSolve.

```
Clear[p,q]
p[x_] =  $\frac{3x^5}{10} - 3x^4 + 11x^3 - 18x^2 + 12x + 1$ ;
q[x_] =  $-4x^3 + 28x^2 - 56x + 32$ ;
Plot[Tooltip[{p[x], q[x]}], {x, -1, 5}, PlotStyle -> {Black, Gray}]
```

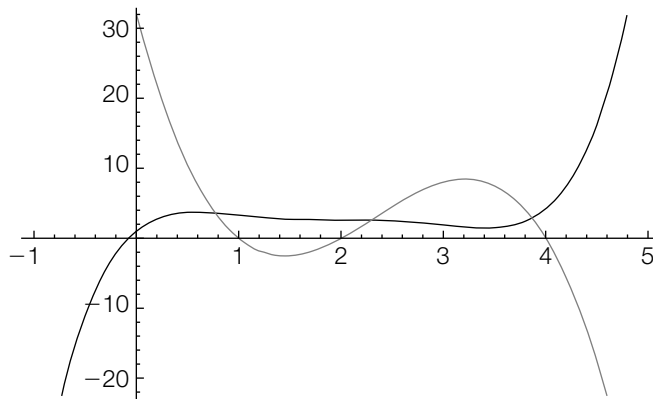


FIGURE 3.36

p and q on the interval $[-1, 5]$.

```
intpts = NRoots[p[x]==q[x], x]
```

```
x==0.772058||x==1.5355-3.57094i||x==1.5355+3.57094i||
```

```
x==2.29182||x==3.86513
```

The numbers are extracted from the list with **Part** (`{...}`). For example, 0.772058 is the second part of the first part of **intpts**. Counting from left to right, 2.29182 is the second part of the fourth part of **intpts**.

```
x1 = intpts[[1, 2]]
```

```
x2 = intpts[[4, 2]]
```

```
x3 = intpts[[5, 2]]
```

```
0.772058
```

```
2.29182
```

```
3.86513
```

Using the roots to the equation $p(x) = q(x)$ and the graph we see that $p(x) \geq q(x)$ for $0.772 \leq x \leq 2.292$ and $q(x) \geq p(x)$ for $2.292 \leq x \leq 3.865$. Hence, an approximation of the area bounded by p and q is given by the sum

$$\int_{0.772}^{2.292} [p(x) - q(x)] dx + \int_{2.292}^{3.865} [q(x) - p(x)] dx.$$

These two integrals are computed with **Integrate** and **NIntegrate**. As expected, the two values are the same.

```
Integrate[p[x]-q[x], {x, x1, x2}] + Integrate[q[x]-p[x], {x, x2, x3}]
```

```
12.1951
```

```
NIntegrate[p[x]-q[x], {x, x1, x2}] + NIntegrate[q[x]-p[x], {x, x2, x3}]
```

```
12.1951
```

We conclude that the area is approximately 12.195.

Graphically, y is a function of x , $y = y(x)$, if the graph of $y = y(x)$ passes the vertical line test.

Parametric Equations

If the curve, C , defined parametrically by $x = x(t)$, $y = y(t)$, $a \leq t \leq b$ is a nonnegative continuous function of x and $x(a) < x(b)$, the area under the graph of C and above the x -axis is

$$\int_{x(a)}^{x(b)} y dx = \int_a^b y(t)x'(t)dt.$$

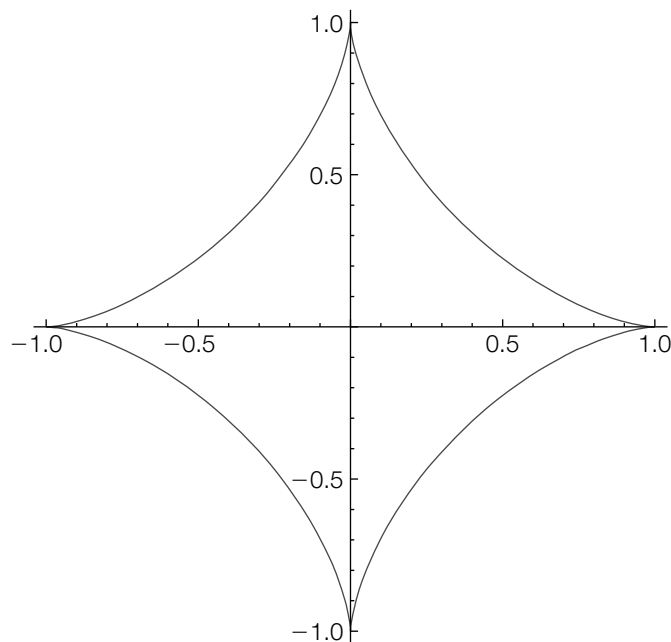
Example 3.3.8 (The Astroid). Find the area enclosed by the **astroid** $x = \sin^3 t$, $y = \cos^3 t$, $0 \leq t \leq 2\pi$.

Solution We begin by defining x and y and then graphing the astroid with **ParametricPlot** in Figure 3.37.

```
x[t_] = Sin[t]^3;
```

```
y[t_] = Cos[t]^3;
```

```
ParametricPlot[{x[t], y[t]}, {t, 0, 2Pi}, AspectRatio->Automatic]
```

**FIGURE 3.37**

The astroid $x = \sin^3 t$, $y = \cos^3 t$, $0 \leq t \leq 2\pi$.

Observe that $x(0) = 0$ and $x(\pi/2) = 1$, and the graph of the astroid in the first quadrant is given by $x = \sin^3 t$, $y = \cos^3 t$, $0 \leq t \leq \pi/2$. Hence, the area of the astroid in the first quadrant is given by

$$\int_0^{\pi/2} y(t)x'(t) dt = 3 \int_0^{\pi/2} \sin^2 t \cos^4 t dt$$

and the total area is given by

$$A = 4 \int_0^{\pi/2} y(t)x'(t) dt = 12 \int_0^{\pi/2} \sin^2 t \cos^4 t dt = \frac{3}{8} \pi \approx 1.178,$$

which is computed with `Integrate` and then approximated with `N`.

```
area = 4Integrate[y[t]x'[t], {t, 0, Pi/2}]
```

```
 $\frac{3\pi}{8}$ 
```

```
N[area]
```

```
1.1781
```

Polar Coordinates

For problems involving “circular symmetry,” it is often easier to work in polar coordinates. The relationship between (x, y) in rectangular coordinates and (r, θ) in polar coordinates is given by

$$x = r \cos \theta \quad y = r \sin \theta$$

and

$$r^2 = x^2 + y^2 \quad \tan \theta = \frac{y}{x}.$$

If $r = f(\theta)$ is continuous and nonnegative for $\alpha \leq \theta \leq \beta$, then the **area** A of the region enclosed by the graphs of $r = f(\theta)$, $\theta = \alpha$, and $\theta = \beta$ is

$$A = \frac{1}{2} \int_{\alpha}^{\beta} [f(\theta)]^2 d\theta = \frac{1}{2} \int_{\alpha}^{\beta} r^2 d\theta.$$

Example 3.3.9 (Lemniscate of Bernoulli). The **lemniscate of Bernoulli** is given by

$$(x^2 + y^2)^2 = a^2 (x^2 - y^2),$$

where a is a constant. (a) Graph the lemniscate of Bernoulli if $a = 2$. (b) Find the area of the region bounded by the lemniscate of Bernoulli.

Solution This problem is much easier solved in polar coordinates, so we first convert the equation from rectangular to polar coordinates with **ReplaceAll** (/.) and then solve for r with **Solve**.

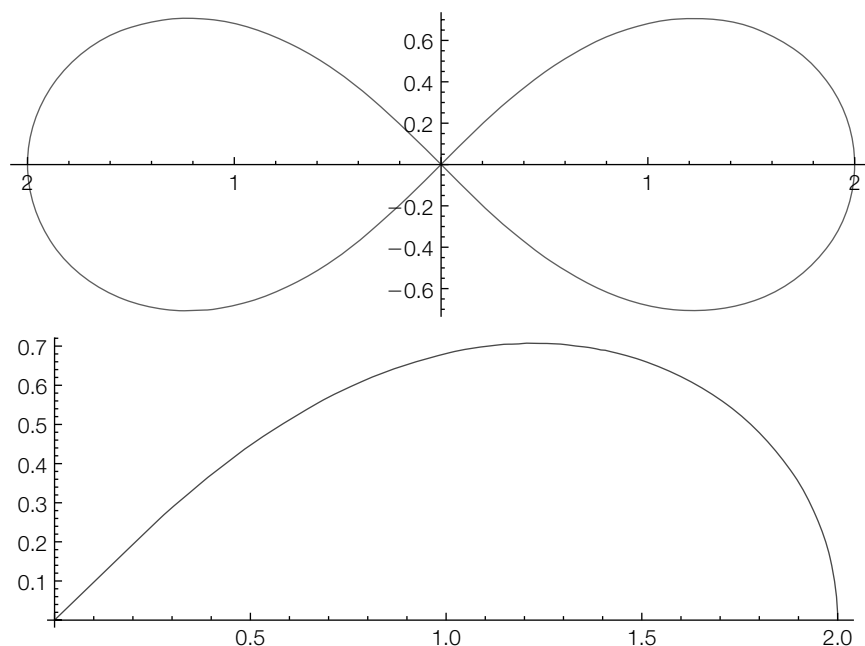
```
lofb = (x^2 + y^2)^2 == a^2 (x^2 - y^2);
topolar = lofb /. {x -> r Cos[t], y -> r Sin[t]}
(r^2 Cos[t]^2 + r^2 Sin[t]^2)^2 == a^2 (r^2 Cos[t]^2 - r^2 Sin[t]^2)
Solve[topolar, r] // Simplify
{ {r -> 0}, {r -> 0}, {r -> -sqrt[a^2 Cos[2t]]}, {r -> sqrt[a^2 Cos[2t]]} }
```

These results indicate that an equation of the lemniscate in polar coordinates is $r^2 = a^2 \cos 2\theta$. The graph of the lemniscate is then generated in Figure 3.38 (top) using **PolarPlot**. The portion of the lemniscate in quadrant one is obtained by graphing $r = 2 \cos 2\theta$, $0 \leq \theta \leq \pi/4$.

```
p1 = PolarPlot[{-2Sqrt[Cos[2t]], 2Sqrt[Cos[2t]]}, {t, 0, 2Pi}];
p2 = PolarPlot[2Sqrt[Cos[2t]], {t, 0, Pi/4}];
Show[GraphicsColumn[{p1, p2}]]
```

Then, taking advantage of symmetry, the area of the lemniscate is given by

$$A = 2 \cdot \frac{1}{2} \int_{-\pi/4}^{\pi/4} r^2 d\theta = 2 \int_0^{\pi/4} r^2 d\theta = 2 \int_0^{\pi/4} a^2 \cos 2\theta d\theta = a^2,$$

**FIGURE 3.38**

(Top) The lemniscate. (Bottom) The portion of the lemniscate in quadrant 1

which we calculate with `Integrate`.

```
Integrate[2a^2 Cos[2t], {t, 0, Pi/4}]
a^2
```

3.3.5 Arc Length

Let $y = f(x)$ be a function for which $f'(x)$ is continuous on an interval $[a, b]$. Then the **arc length** of the graph of $y = f(x)$ from $x = a$ to $x = b$ is given by

$$L = \int_a^b \sqrt{\left(\frac{dy}{dx}\right)^2 + 1} dx. \quad (3.17)$$

The resulting definite integrals used for determining arc length are usually difficult to compute because they involve a radical. In these situations, Mathematica is helpful with approximating solutions to these types of problems.

Example 3.3.10 Find the length of the graph of $y = \frac{x^4}{8} + \frac{1}{4x^2}$ from (a) $x = 1$ to $x = 2$ and from (b) $x = -2$ to $x = -1$.

Solution With no restrictions on the value of x , $\sqrt{x^2} = |x|$. Generally, Mathematica does not automatically algebraically simplify $\sqrt{(dy/dx)^2 + 1}$ because Mathematica does not know if x is positive or negative.

$$y[x_] = x^4/8 + 1/(4x^2);$$

$$i1 = \text{Factor}[y'[x]^2 + 1]$$

$$\frac{(1+x^2)^2(1-x^2+x^4)^2}{4x^6}$$

$$i2 = \text{PowerExpand}[\text{Sqrt}[i1]]$$

$$\frac{(1+x^2)(1-x^2+x^4)}{2x^3}$$

PowerExpand[expr] simplifies radicals in the expression expr assuming that all variables are positive.

In fact, for (b), x is negative so $\frac{1}{2} \sqrt{\frac{(x^6+1)^2}{x^6}} = -\frac{1}{2} \frac{x^6+1}{x^3}$. Mathematica simplifies

$\frac{1}{2} \sqrt{\frac{(x^6+1)^2}{x^6}} = \frac{1}{2} \frac{x^6+1}{x^3}$ and correctly evaluates the arc length integral (3.17) for (a).

$$\text{Integrate}[\text{Sqrt}[y'[x]^2 + 1], \{x, 1, 2\}]$$

$$\frac{33}{16}$$

For (b), we compute the arc length integral (3.17).

$$\text{Integrate}[\text{Sqrt}[y'[x]^2 + 1], \{x, -2, -1\}]$$

$$\frac{33}{16}$$

As we expect, both values are the same.

C is **smooth** if both $x'(t)$ and $y'(t)$ are continuous on (a, b) and not simultaneously zero for $t \in (a, b)$.

Parametric Equations

If the smooth curve, C , defined parametrically by $x = x(t)$, $y = y(t)$, $t \in [a, b]$ is traversed exactly once as t increases from $t = a$ to $t = b$, the arc length of C is given by

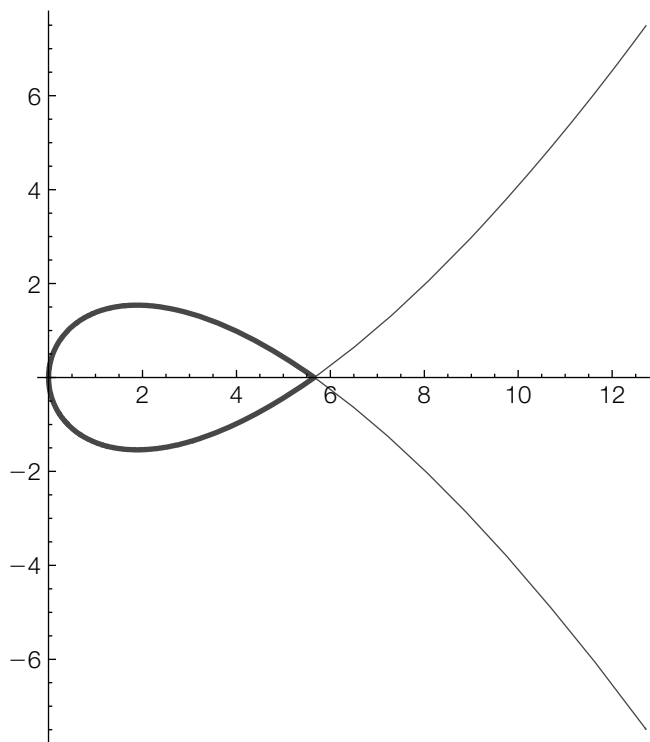
$$L = \int_a^b \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt. \quad (3.18)$$

Example 3.3.11 Find the length of the graph of $x = \sqrt{2}t^2$, $y = 2t - \frac{1}{2}t^3$, $-2 \leq t \leq 2$.

Solution For illustrative purposes, we graph $x = \sqrt{2}t^2$, $y = 2t - \frac{1}{2}t^3$ for $-3 \leq t \leq 3$ and $-2 \leq t \leq 2$ (thickened) in Figure 3.39.

$$x[t_] = t^2 \text{Sqrt}[2]; y[t_] = 2t - 1/2 t^3;$$

$$p1 = \text{ParametricPlot}[\{x[t], y[t]\}, \{t, -3, 3\}];$$

**FIGURE 3.39**

$$x = \sqrt{2}t^2, y = 2t - \frac{1}{2}t^3$$

```
p2 = ParametricPlot[{x[t], y[t]}, {t, -2, 2}, PlotStyle -> Thickness[.01];  
Show[p1, p2, PlotRange -> All]
```

Mathematica is able to compute the exact value of the arc length (3.18), although the result is quite complicated. For length considerations, the result of entering the `i1` command is not displayed here.

```
Factor[x'[t]^2 + y'[t]^2]  
 $\frac{1}{4} (4 - 4t + 3t^2) (4 + 4t + 3t^2)$ 
```

```
i1 = Integrate[2 Sqrt[x'[t]^2 + y'[t]^2], {t, 0, 2}]
```

A more meaningful approximation is obtained with `N` or using `NIntegrate`.

```
N[i1]  
13.7099 + 0.i  
NIntegrate[2 Sqrt[x'[t]^2 + y'[t]^2], {t, 0, 2}]  
13.7099
```

We conclude that the arc length is approximately 13.71.

Polar Coordinates

If the smooth polar curve C given by $r = f(\theta)$, $\alpha \leq \theta \leq \beta$ is traversed exactly once as θ increases from α to β , the arc length of C is given by

$$L = \int_{\alpha}^{\beta} \sqrt{\left(\frac{dr}{d\theta}\right)^2 + r^2} d\theta \quad (3.19)$$

Example 3.3.12 Find the length of the graph of $r = \theta$, $0 \leq \theta \leq 10\pi$.

Solution We begin by defining r and then graphing r with `PolarPlot` in Figure 3.40.

```
r[t_] = t;
PolarPlot[r[t], {t, 0, 10Pi}, AspectRatio->Automatic]
```

Using (3.19), the length of the graph of r is given by $\int_0^{10\pi} \sqrt{1 + \theta^2} d\theta$. The exact value is computed with `Integrate`

```
ev = Integrate[Sqrt[r'[t]^2 + r[t]^2], {t, 0, 10Pi}]
5π√1 + 100π² + ½ArcSinh[10π]
```

and then approximated with `N`.

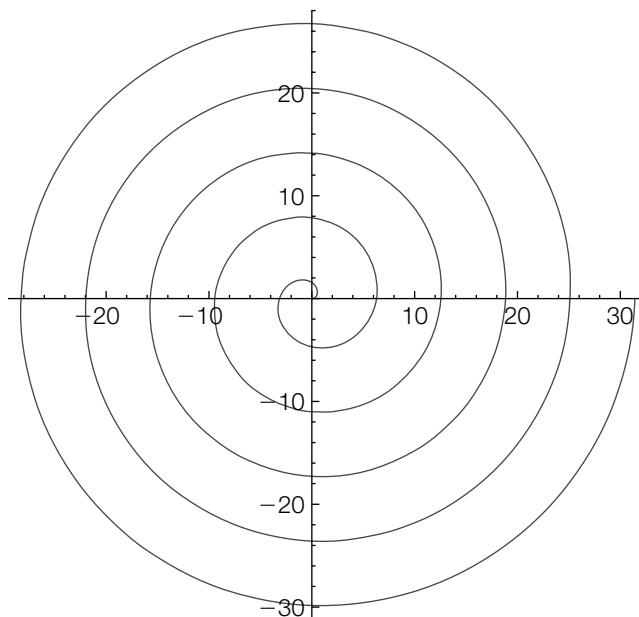


FIGURE 3.40

$r = \theta$ for $0 \leq \theta \leq 10\pi$

N[ev]

495.801

We conclude that the length of the graph is approximately 495.8.

3.3.6 Solids of Revolution

Volume

Let $y = f(x)$ be a nonnegative continuous function on $[a, b]$. The **volume** of the solid of revolution obtained by revolving the region bounded by the graphs of $y = f(x)$, $x = a$, $x = b$, and the x -axis about the x -axis is given by

$$V = \pi \int_a^b [f(x)]^2 dx. \quad (3.20)$$

If $0 \leq a < b$, the **volume** of the solid of revolution obtained by revolving the region bounded by the graphs of $y = f(x)$, $x = a$, $x = b$, and the x -axis about the y -axis is given by

$$V = 2\pi \int_a^b xf(x) dx. \quad (3.21)$$

Example 3.3.13 Let $g(x) = x \sin^2 x$. Find the volume of the solid obtained by revolving the region bounded by the graphs of $y = g(x)$, $x = 0$, $x = \pi$, and the x -axis about (a) the x -axis and (b) the y -axis.

Solution After defining g , we graph g on the interval $[0, \pi]$ in Figure 3.41(a).

With Mathematica 6, for three-dimensional graphics, you can adjust the viewpoint by clicking on the three-dimensional graphics object and dragging to the desired viewing angle.

```
g[x_] = xSin[x]^2;
p1 = Plot[g[x], {x, 0, Pi}, AspectRatio -> Automatic];
```

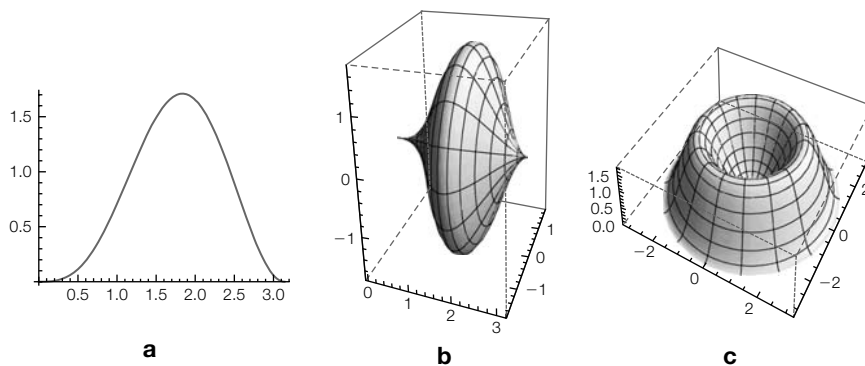


FIGURE 3.41

(a) $g(x)$ for $0 \leq x \leq \pi$. (b) $g(x)$ revolved about the x -axis. (c) $g(x)$ revolved about the y -axis

The volume of the solid obtained by revolving the region about the x -axis is given by (3.20), whereas the volume of the solid obtained by revolving the region about the y -axis is given by (3.21). These integrals are computed with **Integrate** and named **xvol** and **yvol**, respectively. We use **N** to approximate each volume.

```
xvol = Integrate[Pt g[x]^2, {x, 0, Pt}]
```

```
N[xvol]
```

```
 $\frac{1}{64}\pi^2(-15 + 8\pi^2)$   
9.86295
```

```
yvol = Integrate[2 Pt x g[x], {x, 0, Pt}]
```

```
N[yvol]
```

```
 $\frac{1}{6}\pi^2(-3 + 2\pi^2)$   
27.5349
```

We can use **ParametricPlot3D** to visualize the resulting solids by parametrically graphing the equations given by

$$\begin{cases} x = r \cos t \\ y = r \sin t \\ z = g(r) \end{cases}$$

for r between 0 and π and t between $-\pi$ and π to visualize the graph of the solid obtained by revolving the region about the y -axis and by parametrically graphing the equations given by

$$\begin{cases} x = r \\ y = g(r) \cos t \\ z = g(r) \sin t \end{cases}$$

for r between 0 and π and t between $-\pi$ and π to visualize the graph of the solid obtained by revolving the region about the x -axis. (See Figures 3.41(b) and 3.41(c).) In this case, we identify the z -axis as the y -axis. Notice that we are simply using polar coordinates for the x - and y -coordinates, and the height above the x,y -plane is given by $z = g(r)$ because r is replacing x in the new coordinate system.

```
p2 = ParametricPlot3D[{r, g[r]Cos[t], g[r]Sin[t]}, {r, 0, Pt},  
  {t, 0, 2Pt}, PlotPoints -> {30, 30}];
```

```
p3 = ParametricPlot3D[{r Cos[t], r Sin[t], g[r]}, {r, 0, Pt}, {t, 0, 2Pt},  
  PlotPoints -> {30, 30}];
```

p1, **p2**, and **p3** are shown together side-by-side in Figure 3.41 using **Show** together with **GraphicsRow**.

```
Show[GraphicsRow[{p1, p2, p3}]]
```

We now demonstrate a volume problem that requires the method of disks.

Example 3.3.14 Let $f(x) = e^{-(x-3)^2 \cos[4(x-3)]}$. Approximate the volume of the solid obtained by revolving the region bounded by the graphs of $y = f(x)$, $x = 1$, $x = 5$, and the x -axis about the x -axis.

Solution Proceeding as in the previous example, we first define and graph f on the interval $[1, 5]$ in Figure 3.42(a).

```
f[x_] = Exp[-(x-3)^2 Cos[4(x-3)]];
p1 = Plot[f[x], {x, 1, 5}, AspectRatio->Automatic];
```

In this case, an approximation is desired so we use **NIntegrate** to approximate the integral $V = \int_1^5 \pi [f(x)]^2 dx$.

```
NIntegrate[Pi f[x]^2, {x, 1, 5}]
16.0762
```

In the same manner as before, **ParametricPlot3D** can be used to visualize the resulting solid by graphing the set of equations given parametrically by

$$\begin{cases} x = r \\ y = f(r) \cos t \\ z = f(r) \sin t \end{cases}$$

for r between 1 and 5 and t between 0 and 2π . In this case, polar coordinates are used in the y, z -plane, with the distance from the x -axis given by $f(x)$. Because r replaces x in the new coordinate system, $f(x)$ becomes $f(r)$ in these equations. See Figure 3.42(b).

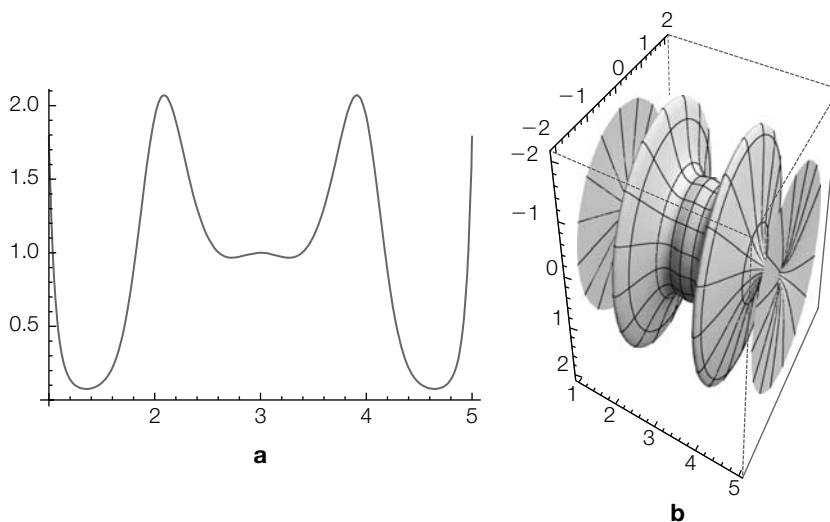


FIGURE 3.42

(a) $f(x)$ for $1 \leq x \leq 5$. (b) $f(x)$ revolved about the x -axis

```
p2 = ParametricPlot3D[{r, f[r]Cos[t], f[r]Sin[t]}, {r, 1, 5},
  {t, 0, 2Pi}, PlotPoints -> {45, 35}];

Show[GraphicsRow[{p1, p2}]]
```

When revolving a curve about the y -axis, you can use `RevolutionPlot3D` rather than the parametrization given previously.

Example 3.3.15 Let $f(x) = \exp(-2(x-2)^2) + \exp(-(x-4)^2)$ for $0 \leq x \leq 6$. Find the minimum and maximum values of $f(x)$ on $[0, 6]$.

Solution (a) Although `Maximize` and `Minimize` cannot find the exact maximum and minimum values, using `N` or `NMaximize` and `NMinimize` gives accurate approximations.

`NMaximize` and `NMinimize` work in the same way as `Maximize` and `Minimize` but return approximations rather than exact results.

```
f[x_] = Exp[-2(x-2)^2] + Exp[-(x-4)^2];
```

```
Maximize[f[x], x]
```

```
Maximize[E^(-4 + x^2) + e^(-2(-2 + x)^2), x]
```

```
Maximize[f[x], x]//N
```

```
{1.01903, {x -> 2.01962}}
```

```
NMaximize[{f[x], 0 <= x <= 6}, x]
```

```
{1.00034, {x -> 3.99864}}
```

```
Minimize[{f[x], 0 <= x <= 6}, x]
```

```
Minimize[{E^(-4 + x^2) + e^(-2(-2 + x)^2), 0 <= x <= 6}, x]
```

```
Minimize[{f[x], 0 <= x <= 6}, x]//N
```

```
{0.495486, {x -> 2.92167}}
```

```
NMinimize[{f[x], 0 <= x <= 6}, x]//N
```

```
{0.495486, {x -> 2.92167}}
```

We double check these results by graphing $f(x)$ and $f'(x)$ in Figure 3.43 and then using `FindRoot` to approximate the critical numbers.

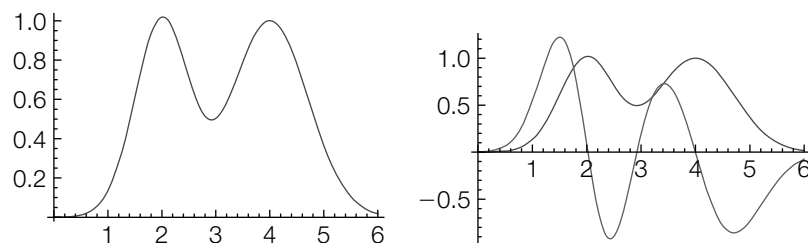


FIGURE 3.43

We use the graph of $f'(x)$ to help us estimate the initial values to approximate the critical numbers with `FindRoot`

```

pf1 = Plot[f[x], {x, 0, 6}]
pf2 = Plot[Tooltip[{f[x], f'[x]}], {x, 0, 6}]
Show[GraphicsRow[{pf1, pf2}]]
Map[FindRoot[f'[x]==0, {x, #1}]]&, {2, 3, 4}]
{{x → 2.01962}, {x → 2.92167}, {x → 3.99864}}

```

(b) Mathematica finds the exact volume of the solids although the results are expressed in terms of the **Error function**, Erf.

```

Integrate[Pi x f[x], {x, 0, 6}]

$$\frac{1}{4e^{32}} \pi (-1 + 2e^{16} + e^{24} - 2e^{28} + 2e^{32} \sqrt{\pi} (4\text{Erf}[2] + 4\text{Erf}[4] + \sqrt{2}(\text{Erf}[2\sqrt{2}] + \text{Erf}[4\sqrt{2}])))$$


```

```

NIntegrate[Pi x f[x], {x, 0, 6}]
30.0673

```

```

Integrate[Pi f[x]^2, {x, 0, 6}]

$$\frac{1}{12e^{8/3}} \pi^{3/2} (3e^{8/3} (\text{Erf}[4] + \text{Erf}[8] + \sqrt{2}(\text{Erf}[2\sqrt{2}] + \text{Erf}[4\sqrt{2}])) + 4\sqrt{3}(\text{Erf}[\frac{8}{\sqrt{3}}] + \text{Erf}[\frac{10}{\sqrt{3}}]))$$


```

```

NIntegrate[Pi f[x]^2, {x, 0, 6}]
7.1682

```

To visualize the solid revolved about the y -axis, we use **RevolutionPlot3D** in p1. We generate the curve in p2, a set of axes, and a representative “slice” of the curve. See Figure 3.44(a). Next, we show the solid together with a representative shell. See Figure 3.44(b).

```

p1 = RevolutionPlot3D[f[x], {x, 1, 5},
  BoxRatios → {2, 2, 1}, PlotRange → {{-5, 5}, {-5, 5}, {0, 5/4}},
  Mesh → None, PlotStyle → Opacity[.4],
  ColorFunction → “LightTemperatureMap”];
p2 = ParametricPlot3D[{x, 0, Exp[-2(x-2)^2] + Exp[-(x-4)^2]},
  {x, 1, 5}, {t, 0, 2Pi}],

```

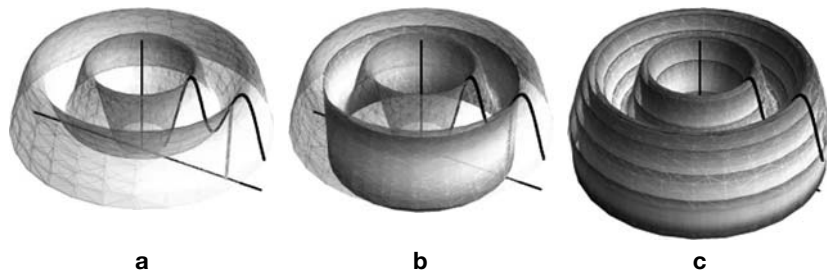


FIGURE 3.44

(a) The solid. (b) The solid with a “typical” shell. (c) Several shells

```

PlotStyle → Thickness[.05], BoxRatios → {2, 2, 1},
Axes → Automatic, Boxed → False];
p3 = ParametricPlot3D[{x, 0, 0}, {x, -5, 5}, {t, 0, 2Pi},
PlotStyle → {Gray, Thickness[.075]}, BoxRatios → {2, 2, 1},
Axes → Automatic, Boxed → False];
p4 = ParametricPlot3D[{0, 0, x}, {x, 0, 5/4}, {t, 0, 2Pi},
PlotStyle → {Gray, Thickness[.1]}, BoxRatios → {2, 2, 1},
Axes → Automatic, Boxed → False];
p5 = Graphics3D[{Gray, Thickness[.01], Line[{{3.6, 0, 0},
{3.6, 0, Exp[-2(3.6-2)^2] + Exp[-(3.6-4)^2]}]}];
p6 = ParametricPlot3D[{3.6Cos[t], 3.6Sin[t], z}, {t, 0, 2Pi},
{z, 0, Exp[-2(3.6-2)^2] + Exp[-(3.6-4)^2]}, Mesh → None,
PlotStyle → Opacity[.8], ColorFunction → "TemperatureMap"];
g1 = Show[p1, p2, p3, p4, p5, Boxed → False, Axes → None]
g2 = Show[p1, p2, p3, p4, p5, p6, Boxed → False, Axes → None]

```

Finally, we show the solid together with several shells in Figure 3.44(c).

```

p7 = Table[ParametricPlot3D[{sp[[i]]Cos[t], sp[[i]]Sin[t], z},
{t, 0, 2Pi},
{z, 0, Exp[-2(sp[[i]]-2)^2] + Exp[-(sp[[i]]-4)^2]},
Mesh → None,
PlotStyle → Opacity[.8],
ColorFunction → "TemperatureMap"], {i, 1, Length[sp]};
g3 = Show[p1, p2, p3, p4, p5, p7, Boxed → False, Axes → None]
Show[GraphicsRow[{g1, g2, g3}]]

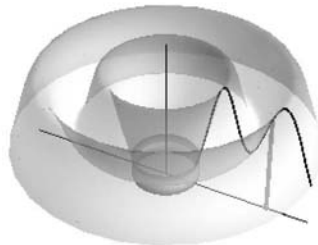
```

With a Do loop you can generate an animation of the process.

```

Do[p8 = ParametricPlot3D[{i Cos[t], i Sin[t], z}, {t, 0, 2 Pi},
{z, 0, Exp[-2 (i-2)^2] + Exp[-(i-4)^2]}, Mesh → None,
PlotStyle → Opacity[.8], ColorFunction → "TemperatureMap"];
Print[Show[p1, p2, p3, p4, p5, p8, Boxed → False, Axes → None]],
{i, 1.1, 4.9, 3.8/29}]

```



For revolving $f(x)$ about the x -axis, we proceed in much the same way. First, we plot $f(x)$ with a set of axes in space.

```
f[x_] = Exp[-2(x-2)^2] + Exp[-(x-4)^2];
p1 = ParametricPlot3D[{x, 0, f[x]}, {x, 0, 6}, PlotStyle -> {Thick, Black},
  PlotRange -> {{0, 6}, {-3/2, 3/2}, {-3/2, 3/2}}, BoxRatios -> {1, 1, 1}];
p1b = ParametricPlot3D[{x, 0, -f[x]}, {x, 0, 6}, PlotStyle -> {Thick, Black},
  PlotRange -> {{0, 6}, {-3/2, 3/2}, {-3/2, 3/2}}, BoxRatios -> {1, 1, 1}];
p2 = ParametricPlot3D[{x, 0, 0}, {x, 0, 6}, {t, 0, 2Pi},
  PlotStyle -> {Gray, Thickness[.075]},
  PlotRange -> {{0, 6}, {-3/2, 3/2}, {-3/2, 3/2}}, BoxRatios -> {1, 1, 1}];
p3 = ParametricPlot3D[{0, 0, x}, {x, -3/2, 3/2}, {t, 0, 2Pi},
  PlotStyle -> {Gray, Thickness[.1]},
  PlotRange -> {{0, 6}, {-3/2, 3/2}, {-3/2, 3/2}}, BoxRatios -> {1, 1, 1}];
Show[p1, p1b, p2, p3]
```

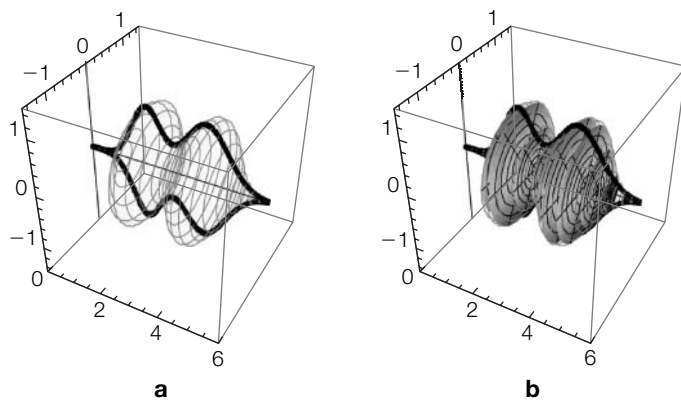
Next, we generate a basic plot of the solid in p4 and then a set of disks inside the solid in t3d.

```
p4 = ParametricPlot3D[{r, f[r]Cos[t], f[r]Sin[t]},
  {r, 0, 6}, {t, 0, 2Pi}, PlotRange -> {{0, 6}, {-3/2, 3/2}, {-3/2, 3/2}},
  BoxRatios -> {1, 1, 1}];
t3d = Table[ParametricPlot3D[{x, rf[x]Cos[t], rf[x]Sin[t]}, {r, 0, 1},
  {t, 0, 2Pi}, PlotRange -> {{0, 6}, {-3/2, 3/2}, {-3/2, 3/2}},
  BoxRatios -> {1, 1, 1}, ColorFunction -> "TemperatureMap", Mesh -> 5],
  {x, 0, 6, 6/14}];
```

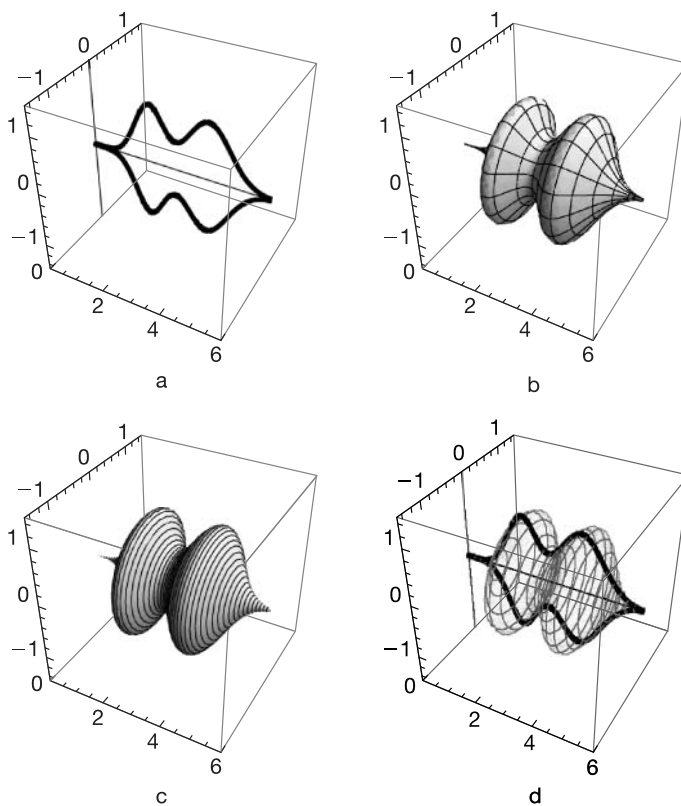
Two variations of the solid are plotted in p5 and p6. In each case, we use `MeshFunctions` to have the contour lines (mesh) correspond to $f(x)$ values rather than the rectangular default mesh. In p6 the solid is made transparent with the `Opacity` option.

```
p5 = ParametricPlot3D[{r, f[r]Cos[t], f[r]Sin[t]},
  {r, 0, 6}, {t, 0, 2Pi}, PlotRange -> {{0, 6}, {-3/2, 3/2}, {-3/2, 3/2}},
  BoxRatios -> {1, 1, 1}, MeshFunctions -> {#1&}, Mesh -> 60];
p6 = ParametricPlot3D[{r, f[r]Cos[t], f[r]Sin[t]},
  {r, 0, 6}, {t, 0, 2Pi}, PlotRange -> {{0, 6}, {-3/2, 3/2},
  {-3/2, 3/2}},
  BoxRatios -> {1, 1, 1}, MeshFunctions -> {#1&}, Mesh -> 25,
  PlotStyle -> Opacity[.2],
  MeshStyle -> {Gray, Thick}];
Show[p1, p1b, p2, p3, p6, t3d]
```

Several combinations of the images are shown in Figures 3.45 and 3.46.

**FIGURE 3.45**

(a) Seeing $f(x)$ on the solid. (b) Disks in the solid

**FIGURE 3.46**

(a) $f(x)$ in space. (b) The basic solid. (c) Contours based on $f(x)$ values. (d) Seeing $f(x)$ on the solid

```
Show[GraphicsRow[{Show[p1, p1b, p2, p3, p6],
  Show[p1, p1b, p2, p3, p6, t3d]}]]
Show[GraphicsGrid[{{Show[p1, p1b, p2, p3], p4},
  {p5, Show[p1, p1b, p2, p3, p6]}]]]
```

To help identify regions, `RegionPlot[constraints, {x, a, b}, {y, a, b}]` attempts to shade the region in the rectangle $[a, b] \times [c, d]$ that satisfies the constraints in constraints.

Example 3.3.16 Let $g(x) = \sqrt{x}$, $h(x) = x^2$, and R be the region bounded by the graphs of $g(x)$ and $h(x)$. Find the volume of the solid obtained by revolving R about (a) the x -axis and (b) the y -axis.

Solution We illustrate the use of `RegionPlot` to help us see R . See Figure 3.47.

```
g[x_] = Sqrt[x];
h[x_] = x^2;
p1a = Plot[{Sqrt[x], x^2}, {x, 0, 2},
  PlotRange -> {{0, 2}, {0, 2}}, AspectRatio -> Automatic]
p1b = RegionPlot[x^2 ≤ y ≤ Sqrt[x], {x, 0, 2}, {y, 0, 2}]
Show[{p1a, p1b}]
Show[GraphicsRow[{p1a, p1b, Show[{p1a, p1b}]}]]
```

We plot the solids with `ParametricPlot3D` and contour lines along the function values using the `MeshFunctions` option in Figure 3.48.

```
p4 = ParametricPlot3D[{r, g[r]Cos[t], g[r]Sin[t]},
  {r, h[r]Cos[t], h[r]Sin[t]},
  {r, 0, 1}, {t, 0, 2Pi}, PlotRange -> {{0, 3/2}, {-5/4, 5/4}, {-5/4, 5/4}},
  BoxRatios -> {1, 1, 1}, MeshFunctions -> {#1&}]
```

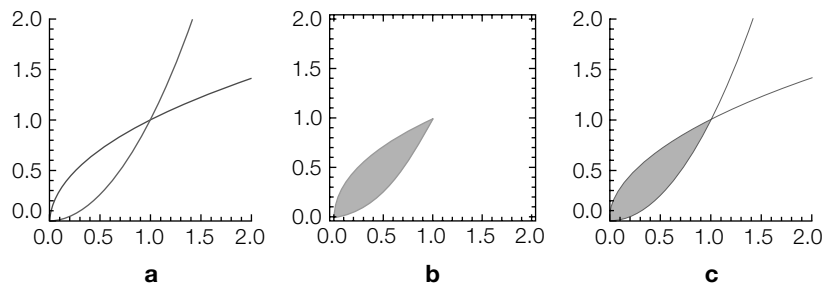


FIGURE 3.47

(a) Graphs of $f(x)$ and $g(x)$. (b) The region in $[0, 2] \times [0, 2]$ for which $x^2 \leq y \leq \sqrt{x}$. (c) The two plots displayed together

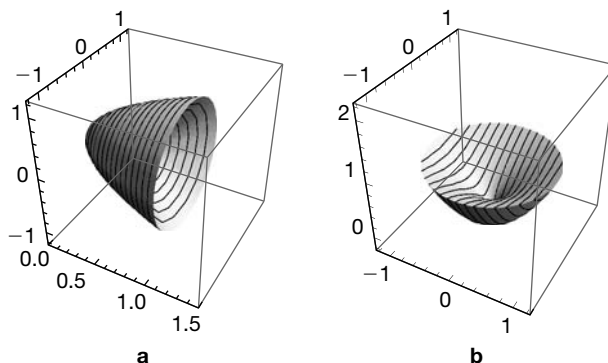


FIGURE 3.48

(a) The solid formed by revolving R about the x -axis. (b) The solid formed by revolving R about the y -axis

```
p5 = ParametricPlot3D[{{rCos[t], rSin[t], g[r]},
  {rCos[t], rSin[t], h[r]}},
  {r, 0, 1}, {t, 0, 2Pi}, PlotRange -> {{-5/4, 5/4}, {-5/4, 5/4}, {-1/4, 9/4}},
  BoxRatios -> {1, 1, 1}, MeshFunctions -> {#1&}]
Show[GraphicsRow[{p4, p5}]]
```

The volume of each solid is then found with `Integrate` and approximated with `N`.

```
Integrate[Pi(g[x]^2 - h[x]^2), {x, 0, 1}]
```

```
 $\frac{3\pi}{10}$ 
```

```
N[%]
```

```
0.942478
```

```
Integrate[Pi(g[x] - h[x]), {x, 0, 1}]
```

```
 $\frac{3\pi}{20}$ 
```

```
N[%]
```

```
0.471239
```

Surface Area

Let $y = f(x)$ be a nonnegative function for which $f'(x)$ is continuous on an interval $[a, b]$. Then the **surface area** of the solid of revolution obtained by revolving the region bounded by the graphs of $y = f(x)$, $x = a$, $x = b$, and the x -axis about the x -axis is given by

$$SA = 2\pi \int_a^b f(x) \sqrt{1 + [f'(x)]^2} dx. \quad (3.22)$$

Example 3.3.17 (Gabriel's Horn). **Gabriel's horn** is the solid of revolution obtained by revolving the area of the region bounded by $y = 1/x$ and the x -axis for $x \geq 1$ about the x -axis. Show that the surface area of Gabriel's horn is infinite but that its volume is finite.

Solution After defining $f(x) = 1/x$, we use `ParametricPlot3D` to visualize a portion of Gabriel's horn in Figure 3.49.

```
f[x_] = 1/x;
ParametricPlot3D[{r, f[r]Cos[t], f[r]Sin[t]}, {r, 1, 10}, {t, 0, 2Pi},
PlotPoints -> {40, 40}, ViewPoint -> {-1.509, -2.739, 1.294}]
```

Using Eq. (3.22), the surface area of Gabriel's horn is given by the improper integral

$$SA = 2\pi \int_1^{\infty} \frac{1}{x} \sqrt{1 + \frac{1}{x^4}} dx = 2\pi \lim_{L \rightarrow \infty} \int_1^L \frac{1}{x} \sqrt{1 + \frac{1}{x^4}} dx.$$

```
step1 = Integrate[2Pi f[x] Sqrt[1 + f'[x]^2], {x, 1, capl}]
```

$$2\pi \text{If} \left[\frac{\text{Im}[\text{capl}]^2}{(-1 + \text{Re}[\text{capl}])^2} \leq 1 \&\& (\text{Re}[\text{capl}] \geq 0 \&\& \text{Im}[\text{capl}] \neq 0), \right.$$

$$\left. \frac{1}{2} \left(\sqrt{2} - \text{ArcSinh}[1] \right) + \frac{\sqrt{1 + \frac{1}{\text{capl}^4}} \left(-\sqrt{1 + \text{capl}^4} + \text{capl}^2 \text{ArcSinh}[\text{capl}^2] \right)}{2\sqrt{1 + \text{capl}^4}}, \right.$$

$$\left. \text{Integrate} \left[\frac{\sqrt{1 + \frac{1}{x^4}}}{x}, \{x, 1, \text{capl}\}, \right. \right.$$

$$\left. \left. \text{Assumptions} \rightarrow \left(\frac{\text{Im}[\text{capl}]^2}{(-1 + \text{Re}[\text{capl}])^2} \leq 1 \&\& (\text{Re}[\text{capl}] \geq 0 \&\& \text{Im}[\text{capl}] \neq 0) \right) \right] \right]$$

```
Limit[step1, capl -> Infinity]
```

∞

On the other hand, using Eq. (3.20) the volume of Gabriel's horn is given by the improper integral

$$V = 2\pi \int_1^{\infty} \frac{1}{x^2} dx = \pi \lim_{L \rightarrow \infty} \int_1^L \frac{1}{x^2} dx,$$

which converges to π .

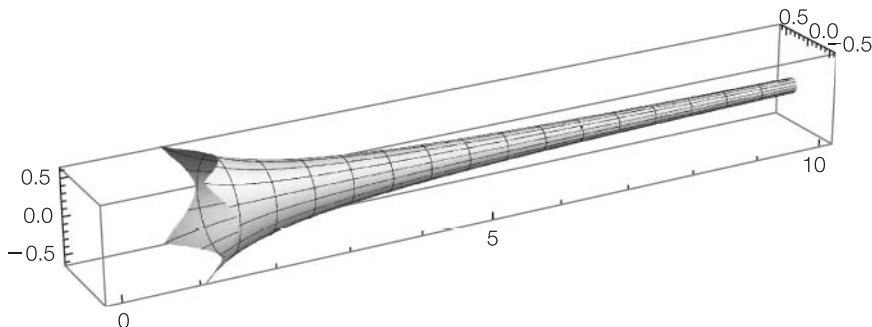


FIGURE 3.49

A portion of Gabriel's horn

```

step1 = Integrate[Pif[x]^2, {x, 1, capl}]
π If [Re[capl] ≥ 0 || Im[capl] ≠ 0, 1 -  $\frac{1}{\text{capl}}$ ,
Integrate [ $\frac{1}{x^2}$ , {x, 1, capl}, Assumptions → !(Re[capl] ≥ 0 || Im[capl] ≠ 0)] ]
Limit[step1, capl -> Infinity]
π
Integrate[Pif[x]^2, {x, 1, Infinity}]
π

```

3.4 SERIES

3.4.1 Introduction to Sequences and Series

Sequences and series are usually discussed in the third quarter or second semester of introductory calculus courses. Most students find that it is one of the most difficult topics covered in calculus. A **sequence** is a function with domain consisting of the positive integers. The **terms** of the sequence $\{a_n\}$ are $a_1, a_2, a_3, \dots$. The n th term is a_n ; the $(n+1)$ st term is a_{n+1} . If $\lim_{n \rightarrow \infty} a_n = L$, we say that $\{a_n\}$ **converges** to L . If $\{a_n\}$ does not converge, $\{a_n\}$ **diverges**. We can sometimes prove that a sequence converges by applying the following theorem.

Theorem 7. *Every bounded monotonic sequence converges.*

A sequence $\{a_n\}$ is monotonic if $\{a_n\}$ is increasing ($a_{n+1} \geq a_n$ for all n) or decreasing ($a_{n+1} \leq a_n$ for all n).

In particular, Theorem 7 gives us the following special cases.

1. If $\{a_n\}$ has positive terms and is eventually decreasing, $\{a_n\}$ converges.
2. If $\{a_n\}$ has negative terms and is eventually increasing $\{a_n\}$ converges.

After you have defined a sequence, use Table to compute the first few terms of the sequence.

1. Table[a[n], {n, 1, m}] returns the list $\{a_1, a_2, a_3, \dots, a_m\}$.
2. Table[a[n], {n, k, m}] returns $\{a_k, a_{k+1}, a_{k+2}, \dots, a_m\}$.

Example 3.4.1 If $a_n = \frac{50^n}{n!}$, show that $\lim_{n \rightarrow \infty} a_n = 0$.

Remark 3.4 An extensive database of integer sequences can be found at the On Line Encyclopedia of Integer Sequences,

<http://www.research.att.com/~njas/sequences/Seis.html>

Solution We remark that the symbol $n!$ in the denominator of a_n represents the **factorial sequence**:

$$n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1.$$

We begin by defining a_n and then computing the first few terms of the sequence with Table.

a[n_] := 50^n/n!;

afewterms = Table[a[n], {n, 1, 10}]

$\{50, 1250, \frac{62500}{3}, \frac{781250}{3}, \frac{7812500}{3}, \frac{195312500}{9}, \frac{9765625000}{63},$
 $\frac{61035156250}{63}, \frac{3051757812500}{567}, \frac{15258789062500}{567}\}$

N[afewterms]

$\{50., 1250., 20833.3, 260417., 2.60417 \times 10^6, 2.17014 \times 10^7,$
 $1.5501 \times 10^8, 9.68812 \times 10^8, 5.38229 \times 10^9, 2.69114 \times 10^{10}\}$

The first few terms increase in magnitude. In fact, this is further confirmed by graphing the first few terms of the sequence with ListPlot in Figure 3.50(a). Based on the graph and the values of the first few terms we might incorrectly conclude that the sequence diverges.

p1 = ListPlot[afewterms];

However, notice that $a_{n+1} = \frac{50}{n+1}a_n \Rightarrow \frac{a_{n+1}}{a_n} = \frac{50}{n+1}$. Because $50/(n+1) < 1$ for $n > 49$, we conclude that the sequence is decreasing for $n > 49$. Because it has positive terms, it is bounded below by 0 so the sequence converges by Theorem 7. Let $L = \lim_{n \rightarrow \infty} a_n$. Then,

$$\begin{aligned}\lim_{n \rightarrow \infty} a_{n+1} &= \lim_{n \rightarrow \infty} \frac{50}{n+1} a_n \\ L &= \lim_{n \rightarrow \infty} \frac{50}{n+1} \cdot L \\ L &= 0.\end{aligned}$$

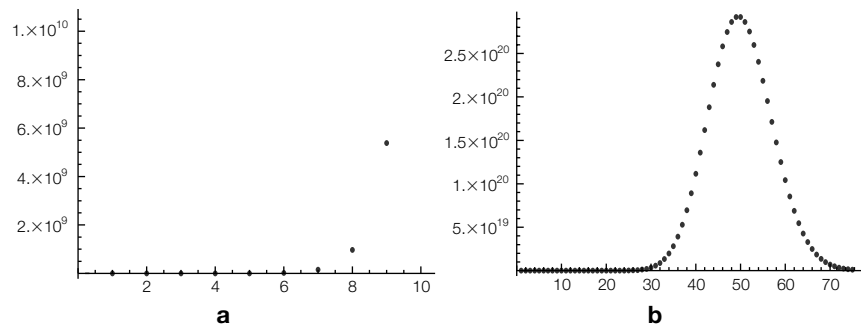


FIGURE 3.50

(a) The first few terms of a_n (b) The first 75 terms of a_n

When we graph a larger number of terms, it is clear that the limit is 0. (See Figure 3.50(b).) It is a good exercise to show that for any real value of x , $\lim_{n \rightarrow \infty} \frac{x^n}{n!} = 0$.

```
p2 = ListPlot[Evaluate[Table[a[k], {k, 1, 75}]]];
Show[GraphicsRow[{p1, p2}]]
```

An **infinite series** is a series of the form

$$\sum_{k=1}^{\infty} a_k, \quad (3.23)$$

where $\{a_n\}$ is a sequence. The n th **partial sum** of (3.23) is

$$s_n = \sum_{k=1}^n a_k = a_1 + a_2 + \cdots + a_n. \quad (3.24)$$

Notice that the partial sums of the series (3.23) form a sequence $\{s_n\}$. Hence, we say that the infinite series (3.23) **converges** to L if the sequence of partial sums $\{s_n\}$ converges to L and write

$$\sum_{k=1}^{\infty} a_k = L.$$

The infinite series (3.23) **diverges** if the sequence of partial sums diverges. Given the infinite series (3.23),

$$\text{Sum}[a[k], \{k, 1, n\}]$$

calculates the n th partial sum (3.24). In *some* cases, if the infinite series (3.23) converges,

$$\text{Sum}[a[k], \{k, 1, \text{Infinity}\}]$$

can compute the value of the infinite sum. In addition to using Sum to compute finite and infinite sums, you can use the  button on the **Basic-MathInput** palette to calculate sums. You should think of the Sum function as a “fragile” command and be certain to carefully examine its results.

Example 3.4.2 Determine whether each series converges or diverges. If the series converges, find its sum. (a) $\sum_{k=1}^{\infty} (-1)^{k+1}$; (b) $\sum_{k=2}^{\infty} \frac{2}{k^2 - 1}$ (c) $\sum_{k=0}^{\infty} ar^k$.

Solution For (a), we compute the n th partial sum (3.24) in sn with Sum.

$$\text{sn} = \text{Sum}[(-1)^{(k+1)}, \{k, 1, n\}]$$

$$\frac{1}{2} (1 - (-1)^n)$$

Notice that the odd partial sums are 1: $s_{2n+1} = \frac{1}{2} ((-1)^{2n+1+1} + 1) = \frac{1}{2}(1 + 1) = 1$, whereas the even partial sums are 0: $s_{2n} = \frac{1}{2} ((-1)^{2n+1} + 1) = \frac{1}{2}(-1 + 1) = 0$. We

confirm that the limit of the partial sums does not exist with **Limit**. Mathematica's result indicates that it cannot determine the limit. The series diverges.

Limit[sn, n → Infinity]

$$\frac{1}{2} (1 - e^{2i \text{Interval}[\{0, \pi\}]})$$

Similarly, when we attempt to compute the infinite sum with **Sum**, Mathematica is able to determine that the partial sums diverge, which means that the infinite series diverges.

Sum[(-1)^(k+1), {k, 1, Infinity}]

Sum::div : Sum does not converge.}}

$$\sum_{k=1}^{\infty} (-1)^{1+k}$$

For (b), we have a *telescoping series*. Using partial fractions,

$$\begin{aligned} \sum_{k=2}^{\infty} \frac{2}{k^2 - 1} &= \sum_{k=2}^{\infty} \left(\frac{1}{k-1} - \frac{1}{k+1} \right) \\ &= \left(1 - \frac{1}{3} \right) + \left(\frac{1}{2} - \frac{1}{4} \right) + \left(\frac{1}{3} - \frac{1}{5} \right) + \cdots + \left(\frac{1}{n-2} - \frac{1}{n} \right) \\ &\quad + \left(\frac{1}{n-1} - \frac{1}{n+1} \right) + \cdots, \end{aligned}$$

we see that the n th partial sum is given by

$$s_n = \frac{3}{2} - \frac{1}{n} - \frac{1}{n+1}$$

and $s_n \rightarrow 3/2$ as $n \rightarrow \infty$ so the series converges to $3/2$:

$$\sum_{k=2}^{\infty} \frac{2}{k^2 - 1} = \frac{3}{2}.$$

We perform the same steps with Mathematica using **Sum**, **Apart**, and **Limit**.

sn = Sum[1/(k-1) - 1/(k+1), {k, 2, n}]

$$\frac{(-1+n)(2+3n)}{2n(1+n)}$$

Apart[sn]

$$\frac{3}{2} - \frac{1}{n} - \frac{1}{1+n}$$

Limit[sn, n → Infinity]

$$\frac{3}{2}$$

(c) A series of the form $\sum_{k=0}^{\infty} ar^k$ is called a **geometric series**. We compute the n th partial sum of the geometric series with **Sum**.

sn = Sum[ar^k, {k, 0, n}]

$$\frac{a(-1+r^{1+n})}{-1+r}$$

Apart computes the partial fraction decomposition of a rational expression.

When using `Limit` to determine the limit of s_n as $n \rightarrow \infty$, we see that Mathematica returns the limit unevaluated because Mathematica does not know the value of r .

Limit[sn, n → Infinity]

$$\text{Limit}\left[\frac{a(-1+r^{1+n})}{-1+r}, n \rightarrow \infty\right]$$

In fact, the geometric series diverges if $|r| \geq 1$ and converges if $|r| < 1$. Observe that if we simply compute the sum with `Sum`, Mathematica returns $a/(1-r)$, which is correct if $|r| < 1$ but incorrect if $|r| \geq 1$.

Sum[ar^k, {k, 0, Infinity}]

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

However, the result of entering

Sum[(-5/3)^k, {k, 0, Infinity}]

`Sum::div : Sumdoesnotconverge. >>`

$$\sum_{k=0}^{\infty} \left(-\frac{5}{3}\right)^k$$

is correct because the series $\sum_{k=0}^{\infty} \left(-\frac{5}{3}\right)^k$ is geometric with $|r| = 5/3 \geq 1$ and, consequently, diverges. Similarly,

Sum[9(1/10)^k, {k, 1, Infinity}]

$$1$$

is correct because $\sum_{k=1}^{\infty} 9 \left(\frac{1}{10}\right)^k$ is geometric with $a = 9/10$ and $r = 1/10$ so the series converges to

$$\frac{a}{1-r} = \frac{9/10}{1-1/10} = 1.$$

3.4.2 Convergence Tests

Frequently used convergence tests are stated in the following theorems.

Theorem 8 (The Divergence Test). Let $\sum_{k=1}^{\infty} a_k$ be an infinite series. If $\lim_{k \rightarrow \infty} a_k \neq 0$, then $\sum_{k=1}^{\infty} a_k$ diverges.

Theorem 9 (The Integral Test). Let $\sum_{k=1}^{\infty} a_k$ be an infinite series with positive terms. If $f(x)$ is a decreasing continuous function for which $f(k) = a_k$ for all k , then $\sum_{k=1}^{\infty} a_k$ and $\int_1^{\infty} f(x) dx$ either both converge or both diverge.

Theorem 10 (The Ratio Test). Let $\sum_{k=1}^{\infty} a_k$ be an infinite series with positive terms and let $\rho = \lim_{k \rightarrow \infty} \frac{a_{k+1}}{a_k}$.

1. If $\rho < 1$, $\sum_{k=1}^{\infty} a_k$ converges.
2. If $\rho > 1$, $\sum_{k=1}^{\infty} a_k$ diverges.
3. If $\rho = 1$, the ratio test is inconclusive.

Theorem 11 (The Root Test). Let $\sum_{k=1}^{\infty} a_k$ be an infinite series with positive terms and let $\rho = \lim_{k \rightarrow \infty} \sqrt[k]{a_k}$.

1. If $\rho < 1$, $\sum_{k=1}^{\infty} a_k$ converges.
2. If $\rho > 1$, $\sum_{k=1}^{\infty} a_k$ diverges.
3. If $\rho = 1$, the root test is inconclusive.

Theorem 12 (The Limit Comparison Test). Let $\sum_{k=1}^{\infty} a_k$ and $\sum_{k=1}^{\infty} b_k$ be infinite series with positive terms and let $L = \lim_{k \rightarrow \infty} \frac{a_k}{b_k}$. If $0 < L < \infty$, then either both series converge or both series diverge.

Example 3.4.3 Determine whether each series converges or diverges. (a) $\sum_{k=1}^{\infty} \left(1 + \frac{1}{k}\right)^k$; (b) $\sum_{k=1}^{\infty} \frac{1}{k^p}$; (c) $\sum_{k=1}^{\infty} \frac{k}{3^k}$; (d) $\sum_{k=1}^{\infty} \frac{(k!)^2}{(2k)!}$; (e) $\sum_{k=1}^{\infty} \left(\frac{k}{4k+1}\right)^k$; and (f) $\sum_{k=1}^{\infty} \frac{2\sqrt{k}+1}{(\sqrt{k}+1)(2k+1)}$.

Solution (a) Using Limit, we see that the limit of the terms is $e \neq 0$ so the series diverges by the divergence test, Theorem 8.

Limit $[(1 + 1/k)^k, k \rightarrow \text{Infinity}]$
e

It is a very good exercise to show that the limit of the terms of the series is e by hand. Let $L = \lim_{k \rightarrow \infty} (1 + 1/k)^k$. Take the logarithm of each side of this equation and apply L'Hôpital's rule:

$$\begin{aligned} \ln L &= \lim_{k \rightarrow \infty} \ln \left(1 + \frac{1}{k}\right)^k \\ \ln L &= \lim_{k \rightarrow \infty} k \ln \left(1 + \frac{1}{k}\right) \\ \ln L &= \lim_{k \rightarrow \infty} \frac{\ln \left(1 + \frac{1}{k}\right)}{\frac{1}{k}} \\ &= \frac{\frac{1}{1 + \frac{1}{k}} \cdot -\frac{1}{k^2}}{-\frac{1}{k^2}} \\ \ln L &= 1. \end{aligned}$$

Exponentiating yields $L = e^{\ln L} = e^1 = e$. (b) A series of the form $\sum_{k=1}^{\infty} \frac{1}{k^p}$ ($p > 0$) is called a **p-series**. Let $f(x) = x^{-p}$. Then, $f(x)$ is continuous and decreasing for $x \geq 1$, $f(k) = k^{-p}$ and

$$\int_1^{\infty} x^{-p} dx = \begin{cases} \infty, & \text{if } p \leq 1 \\ 1/(p-1), & \text{if } p > 1 \end{cases}$$

so the p -series converges if $p > 1$ and diverges if $p \leq 1$. If $p = 1$, the series $\sum_{k=1}^{\infty} \frac{1}{k}$ is called the **harmonic series**.

s1 = Integrate[x^(-p), {x, 1, Infinity}]

If $\left[\operatorname{Re}[p] > 1, \frac{1}{-1+p}, \operatorname{Integrate}[x^{-p}, \{x, 1, \infty\}], \operatorname{Assumptions} \rightarrow \operatorname{Re}[p] \leq 1 \right]$

(c) Let $f(x) = x \cdot 3^{-x}$. Then, $f(k) = k \cdot 3^{-k}$ and $f(x)$ is decreasing for $x > 1/\ln 3$.

f[x_] = x3^(-x);

Factor[f'[x]]

$-3^{-x}(-1 + x \operatorname{Log}[3])$

Solve[-1 + xLog[3]==0, x]

$\left\{ \left\{ x \rightarrow \frac{1}{\operatorname{Log}[3]} \right\} \right\}$

Using **Integrate**, we see that the improper integral $\int_1^{\infty} f(x) dx$ converges.

ival = Integrate[f[x], {x, 1, Infinity}]

N[ival]

$\frac{1 + \operatorname{Log}[3]}{3 \operatorname{Log}[3]^2}$

0.579592

Thus, by the integral test, Theorem 9, we conclude that the series converges. Note that when applying the integral test, if the improper integral converges, its value is *not* the value of the sum of the series. In this case, we see that Mathematica is able to evaluate the sum with **Sum** and the series converges to $3/4$.

Sum[k3^(-k), {k, 1, Infinity}]

$\frac{3}{4}$

(d) If a_k contains factorial functions, the ratio test is a good first test to try. After defining a_k we compute

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{a_{k+1}}{a_k} &= \lim_{k \rightarrow \infty} \frac{\frac{[(k+1)!]^2}{[2(k+1)]}}{(k!)^2} \\ &= \lim_{k \rightarrow \infty} \frac{(k+1)! \cdot (k+1)!}{k! \cdot k!} \frac{(2k)!}{(2k+2)!} \\ &= \lim_{k \rightarrow \infty} \frac{(k+1)^2}{(2k+2)(2k+1)} = \lim_{k \rightarrow \infty} \frac{(k+1)}{2(2k+1)} = \frac{1}{4}. \end{aligned}$$

Because $1/4 < 1$, the series converges by the ratio test. We confirm these results with Mathematica.

Remark 3.5 Use FullSimplify instead of Simplify to simplify expressions involving factorials.

```

a[k_] = (k!)^2/(2k)!;
s1 = FullSimplify[a[k + 1]/a[k]]

$$\frac{1+k}{2+4k}$$

Limit[s1, k → Infinity]

$$\frac{1}{4}$$


```

We illustrate that we can evaluate the sum using Sum and approximate it with N as follows.

```

ev = Sum[a[k], {k, 1, Infinity}]

$$\frac{1}{27} \left( 9 + 2\sqrt{3}\pi \right)$$

N[ev]
0.7364

```

(e) Because

$$\lim_{k \rightarrow \infty} \sqrt[k]{\left(\frac{k}{4k+1}\right)^k} = \lim_{k \rightarrow \infty} \frac{k}{4k+1} = \frac{1}{4} < 1,$$

the series converges by the root test.

```

a[k_] = (k/(4k + 1))^k;
Limit[a[k]^(1/k), k → Infinity]

$$\frac{1}{4}$$


```

As with (d), we can approximate the sum with N and Sum.

```

ev = Sum[a[k], {k, 1, Infinity}]

$$\sum_{k=1}^{\infty} \left(\frac{k}{1+4k}\right)^k$$

N[ev]
0.265757

```

(f) We use the limit comparison test and compare the series to $\sum_{k=1}^{\infty} \frac{\sqrt{k}}{k\sqrt{k}} = \sum_{k=1}^{\infty} \frac{1}{k}$, which diverges because it is a p -series with $p = 1$. Because

$$0 < \lim_{k \rightarrow \infty} \frac{\frac{2\sqrt{k}+1}{(\sqrt{k}+1)(2k+1)}}{\frac{1}{k}} = 1 < \infty$$

and the harmonic series diverges, the series diverges by the limit comparison test.

```

a[k_] = (2Sqrt[k] + 1)/((Sqrt[k] + 1)(2k + 1));
b[k_] = 1/k;
Limit[a[k]/b[k], k → Infinity]
1

```

3.4.3 Alternating Series

An **alternating series** is a series of the form

$$\sum_{k=1}^{\infty} (-1)^k a_k \quad \text{or} \quad \sum_{k=1}^{\infty} (-1)^{k+1} a_k \quad (3.25)$$

where $\{a_k\}$ is a sequence with positive terms.

Theorem 13 (Alternating Series Test). *If $\{a_k\}$ is decreasing and $\lim_{k \rightarrow \infty} a_k = 0$, the alternating series (3.25) converges.*

The alternating series (3.25) **converges absolutely** if $\sum_{k=1}^{\infty} a_k$ converges.

Theorem 14. *If the alternating series (3.25) converges absolutely, it converges.*

If the alternating series (3.25) converges but does not converge absolutely, we say that it **conditionally converges**.

Example 3.4.4 Determine whether each series converges or diverges. If the series converges, determine whether the convergence is conditional or absolute. (a) $\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}$; (b) $\sum_{k=1}^{\infty} (-1)^{k+1} \frac{(k+1)!}{4^k (k!)^2}$; and (c) $\sum_{k=1}^{\infty} (-1)^{k+1} \left(1 + \frac{1}{k}\right)^k$.

Solution (a) Because $\{1/k\}$ is decreasing and $1/k \rightarrow 0$ as $k \rightarrow \infty$, the series converges. The series does not converge absolutely because the harmonic series diverges. Hence, $\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}$, which is called the **alternating harmonic series**, converges conditionally. We see that this series converges to $\ln 2$ with Sum.

```
a[k_] = (-1)^(k+1)/k;
Sum[a[k], {k, 1, Infinity}]
Log[2]
```

(b) We test for absolute convergence first using the ratio test. Because

$$\lim_{k \rightarrow \infty} \frac{\frac{((k+1)+1)!}{4^{k+1} [(k+1)!]^2}}{\frac{(k+1)!}{4^k (k!)^2}} = \lim_{k \rightarrow \infty} \frac{k+2}{4(k+1)^2} = 0 < 1,$$

```
a[k_] = (k+1)!/(4^k k(k!)^2);
s1 = FullSimplify[a[k+1]/a[k]]
Limit[s1, k -> Infinity]
0
```

the series converges absolutely by the ratio test. Absolute convergence implies convergence so the series converges. (c) Because $\lim_{k \rightarrow \infty} \left(1 + \frac{1}{k}\right)^k = e$,

$\lim_{k \rightarrow \infty} (-1)^{k+1} \left(1 + \frac{1}{k}\right)^k$ does not exist, so the series diverges by the divergence test. We confirm that the limit of the terms is not zero with Limit.

Sum $[(-1)^{(k+1)}a[k], \{k, 1, \text{Infinity}\}]$

$$\frac{-3+4e^{1/4}}{4e^{1/4}}$$

a[k_] = $(-1)^{(k+1)}(1 + 1/k)^k$;

Sum $[a[k], \{k, 1, \text{Infinity}\}]$

Sum::div : Sum does not converge.}}

$$\sum_{k=1}^{\infty} (-1)^{1+k} \left(1 + \frac{1}{k}\right)^k$$

Limit $[a[k], k \rightarrow \text{Infinity}]$

$$-e^{2\text{Interval}[\{0, \pi\}]}$$

3.4.4 Power Series

Let x_0 be a number. A **power series** in $x - x_0$ is a series of the form

$$\sum_{k=0}^{\infty} a_k (x - x_0)^k. \quad (3.26)$$

A fundamental problem is determining the values of x , if any, for which the power series converges, the **interval of convergence**.

Theorem 15. *For the power series (3.26), exactly one of the following is true.*

1. *The power series converges absolutely for all values of x . The interval of convergence is $(-\infty, \infty)$.*
2. *There is a positive number r so that the series converges absolutely if $x_0 - r < x < x_0 + r$. The series may or may not converge at $x = x_0 - r$ and $x = x_0 + r$. The interval of convergence will be one of $(x_0 - r, x_0 + r)$, $[x_0 - r, x_0 + r)$, $(x_0 - r, x_0 + r]$, or $[x_0 - r, x_0 + r]$.*
3. *The series converges only if $x = x_0$. The interval of convergence is $\{x_0\}$.*

Example 3.4.5 Determine the interval of convergence for each of the following power series.

(a) $\sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}$; (b) $\sum_{k=0}^{\infty} \frac{k!}{1000^k} (x-1)^k$; and (c) $\sum_{k=1}^{\infty} \frac{2^k}{\sqrt{k}} (x-4)^k$.

Solution (a) We test for absolute convergence first using the ratio test. Because

$$\lim_{k \rightarrow \infty} \left| \frac{\frac{(-1)^{k+1}}{(2(k+1)+1)!} x^{2(k+1)+1}}{\frac{(-1)^k}{(2k+1)!} x^{2k+1}} \right| = \lim_{k \rightarrow \infty} \frac{1}{2(k+1)(2k+3)} x^2 = 0 < 1$$

```

a[x_, k_] = (-1)^k/(2k + 1)!x^(2k + 1);
s1 = FullSimplify[a[x, k + 1]/a[x, k]]
Limit[s1, k → Infinity]

$$-\frac{x^2}{6+10k+4k^2}$$

0

```

for all values of x , we conclude that the series converges absolutely for all values of x ; the interval of convergence is $(-\infty, \infty)$. In fact, we will see later that this series converges to $\sin x$:

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^{k+1}}{(2k+1)!} x^{2k+1} = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \dots,$$

which means that the partial sums of the series converge to $\sin x$. Graphically, we can visualize this by graphing partial sums of the series together with the graph of $y = \sin x$. Note that the partial sums of a series are a recursively defined function: $s_n = s_{n-1} + a_n$, $s_0 = a_0$. We use this observation to define p to be the n th partial sum of the series. We use the form $p[x_, n_] := p[x, n] = \dots$ so that Mathematica remembers the partial sums computed. That is, once $p[x, 3]$ is computed, Mathematica need not recompute $p[x, 3]$ when computing $p[x, 4]$.

In Figure 3.51 (top) we graph $p_n(x) = \sum_{k=0}^n \frac{(-1)^k}{(2k+1)!} x^{2k+1}$ together with $y = \sin x$ for $n = 1, 5$, and 10 . In the graphs, notice that as n increases, the graphs of $p_n(x)$ more closely resemble the graph of $y = \sin x$.

When you use Tooltip, placing the cursor over the plot shows you the function being plotted.

```

Clear[p]
p[x_, 0] = a[x, 0];
p[x_, n_] := p[x, n] = p[x, n - 1] + a[x, n]
p[x, 2]

$$x - \frac{x^3}{6} + \frac{x^5}{120}$$


```

```

p1 = Plot[Tooltip[{Sin[x], p[x, 1], p[x, 5], p[x, 10]}],
{x, -2Pi, 2Pi}, PlotRange → {-Pi, Pi}, AspectRatio → Automatic];

```

We use Manipulate to investigate how n affects the situation with

```

p2 = Manipulate[Plot[Tooltip[{Sin[x], p[x, n]}],
{x, -4Pi, 4Pi}, PlotRange → {-Pi, Pi}, AspectRatio → Automatic],
{{n, 5}, 1, 25, 1}];
Show[GraphicsColumn[{p1, p2}]]

```

(b) As in (a), we test for absolute convergence first using the ratio test:

$$\lim_{k \rightarrow \infty} \left| \frac{\frac{(k+1)k!}{1000^{k+1}}(x-1)^{k+1}}{\frac{k!}{1000^k}(x-1)^k} \right| = \frac{1}{1000}(k+1)|x-1| = \begin{cases} 0, & \text{if } x = 1 \\ \infty, & \text{if } x \neq 1. \end{cases}$$

```

a[x_, k_] = k!/1000^k(x - 1)^k;
s1 = FullSimplify[a[x, k + 1]/a[x, k]]
Limit[s1, k → Infinity]

```

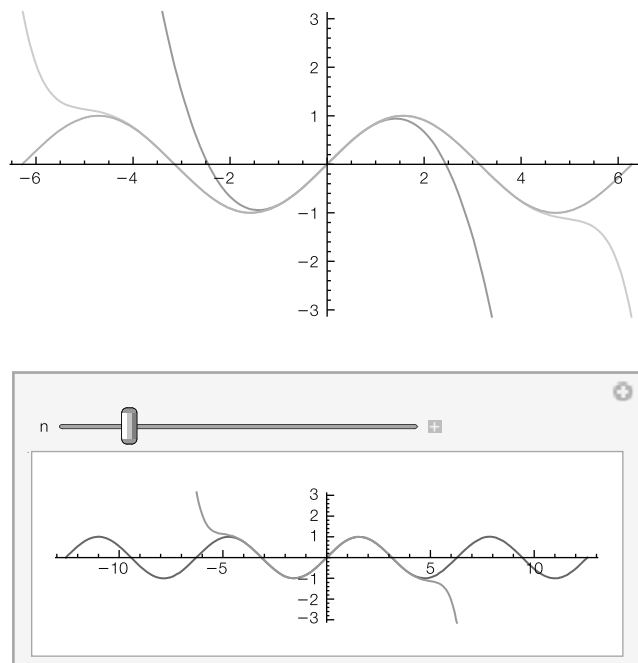


FIGURE 3.51

(Top) $y = \sin x$ together with the graphs of $p_1(x)$, $p_5(x)$, and $p_{10}(x)$. (Bottom) Using Manipulate to investigate the situation

$$\frac{(1+k)(-1+x)}{1000} \\ (-1+x)\infty$$

Be careful of your interpretation of the result of the Limit command because Mathematica does not consider the case $x = 1$ separately: If $x = 1$, the limit is 0. Because $0 < 1$, the series converges by the ratio test.

The series converges only if $x = 1$; the interval of convergence is $\{1\}$. You should observe that if you graph several partial sums for “small” values of n , you might incorrectly conclude that the series converges.

(c) Use the ratio test to check absolute convergence first:

$$\lim_{k \rightarrow \infty} \left| \frac{\frac{2^{k+1}}{\sqrt{k+1}}(x-4)^{k+1}}{\frac{2^k}{\sqrt{k}}(x-4)^k} \right| = \lim_{k \rightarrow \infty} 2\sqrt{\frac{k}{k+1}}|x-4| = 2|x-4|.$$

By the ratio test, the series converges absolutely if $2|x-4| < 1$. We solve this inequality for x with Reduce to see that $2|x-4| < 1$ if $7/2 < x < 9/2$.

```
Clear[a, s1, k]
a[x_, k_] = 2^k k/Sqrt[k](x-4)^k;
```

s1 = Simplify[Abs[a[x, k + 1]/a[x, k]]]

Limit[s1, k → Infinity]

$$2\text{Abs}\left[\sqrt{\frac{k}{1+k}}(-4+x)\right]$$

$$2\text{Abs}[-4+x]$$

Reduce[2Abs[x - 4] < 1, x]

$$\frac{7}{2} < \text{Re}[x] < \frac{9}{2} \&\& -\frac{1}{2}\sqrt{-63 + 32\text{Re}[x] - 4\text{Re}[x]^2} < \text{Im}[x] < \frac{1}{2}\sqrt{-63 + 32\text{Re}[x] - 4\text{Re}[x]^2}$$

From the output, we see that for real values of x , the inequality is satisfied for $7/2 < x < 9/2$. We check $x = 7/2$ and $x = 9/2$ separately. If $x = 7/2$, the series becomes $\sum_{k=1}^{\infty} (-1)^k \frac{1}{\sqrt{k}}$, which converges conditionally.

Simplify[a[x, k]/.x → 7/2]

$$\frac{(-1)^k}{\sqrt{k}}$$

On the other hand, if $x = 9/2$,

Simplify[a[x, k]/.x → 9/2]

$$\frac{1}{\sqrt{k}}$$

the series is $\sum_{k=1}^{\infty} \frac{1}{\sqrt{k}}$, which diverges. We conclude that the interval of convergence is $[7/2, 9/2)$.

3.4.5 Taylor and Maclaurin Series

Let $y = f(x)$ be a function with derivatives of all orders at $x = x_0$. The **Taylor series** for $f(x)$ about $x = x_0$ is

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k. \quad (3.27)$$

The **Maclaurin series** for $f(x)$ is the Taylor series for $f(x)$ about $x = 0$. If $y = f(x)$ has derivatives up to at least order n at $x = x_0$, the n th degree **Taylor polynomial** for $f(x)$ about $x = x_0$ is

$$p_n(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k. \quad (3.28)$$

The n th degree **Maclaurin polynomial** for $f(x)$ is the n th degree Taylor polynomial for $f(x)$ about $x = 0$. Generally, finding Taylor and Maclaurin series using the definition is a tedious task at best.

Example 3.4.6 Find the first few terms of (a) the Maclaurin series and (b) the Taylor series about $x = \pi/4$ for $f(x) = \tan x$.

Solution (a) After defining $f(x) = \tan x$, we use `Table` together with `/.` and `D` to compute $f^{(k)}(0)/k!$ for $k = 0, 1, \dots, 8$.

```
f[x_] = Tan[x];
t1 = Table[{k, D[f[x], {x, k}], D[f[x], {x, k}]/.x -> 0}, {k, 0, 8}];
Short[t1]
{ {0, Tan[x], 0}, {1, Tan[x], 1}, {2, 1, 1}, {3, 0, 0}, {4, -1, -1}, {5, 0, 0}, {6, 1, 1}, {7, 0, 0}, {8, -1, -1} }
```

Use `Short` to obtain an abbreviated result. Many terms will be missing, but with `Short`, you will see the beginning and end of your result.

To see these results in tabular form, enter

```
t1//TableForm
```

For length considerations, the resulting output is not shown here. Another way of approaching the problem is to use `Manipulate`. See Figures 3.52 and 3.53.

```
Manipulate[{k, D[f[x], {x, k}]/FullSimplify, D[f[x], {x, k}]/.x -> 0,
D[f[x], {x, k}]/.x -> 0/N}], {{k, 5}, 0, 25, 1}]
```

Using the values in the table or from the `Manipulate` object, we apply the definition to see that the Maclaurin series is

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k = x + \frac{1}{3}x^3 + \frac{2}{15}x^5 + \frac{17}{315}x^7 + \dots$$

For (b), we repeat (a) using $x = \pi/4$ instead of $x = 0$

```
Manipulate[{k, D[f[x], {x, k}]/FullSimplify, D[f[x], {x, k}]/.x -> Pi/4,
D[f[x], {x, k}]/.x -> Pi/4/N}], {{k, 1}, 0, 25, 1}]
```

and then apply the definition to see that the Taylor series about $x = \pi/4$ is

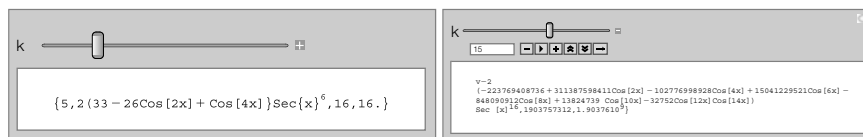


FIGURE 3.52

With `Manipulate`, we can adjust the function and function values



FIGURE 3.53

We use `Manipulate` to investigate series for the tangent function

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k = 1 + 2 \left(x - \frac{\pi}{4}\right) + 2 \left(x - \frac{\pi}{4}\right)^2 + \frac{8}{3} \left(x - \frac{\pi}{4}\right)^3 + \frac{10}{3} \left(x - \frac{\pi}{4}\right)^4 + \frac{64}{15} \left(x - \frac{\pi}{4}\right)^5 + \frac{244}{45} \left(x - \frac{\pi}{4}\right)^6 + \dots$$

From the series, we can see various Taylor and Maclaurin polynomials. For example, the third Maclaurin polynomial is

$$p_3(x) = x + \frac{1}{3}x^3$$

and the fourth degree Taylor polynomial about $x = \pi/4$ is

$$p_4(x) = 1 + 2 \left(x - \frac{\pi}{4}\right) + 2 \left(x - \frac{\pi}{4}\right)^2 + \frac{8}{3} \left(x - \frac{\pi}{4}\right)^3 + \frac{10}{3} \left(x - \frac{\pi}{4}\right)^4.$$

The command `Series[f[x], {x, x0, n}]` computes (3.27) to (at least) order $n - 1$. Because of the O -term in the result that represents the terms that are omitted from the power series for $f(x)$ expanded about the point $x = x_0$, the result of entering a `Series` command is not a function that can be evaluated if x is a particular number. We remove the remainder (O -) term of the power series `Series[f[x], {x, x0, n}]` with the command `Normal` and can then evaluate the resulting polynomial for particular values of x .

Example 3.4.7 Find the first few terms of the Taylor series for $f(x)$ about $x = x_0$. (a) $f(x) = \cos x$, $x = 0$; (b) $f(x) = 1/x^2$, $x = 1$.

Solution Entering

Series[Cos[x], {x, 0, 4}]

$$1 - \frac{x^2}{2} + \frac{x^4}{24} + O[x]^5$$

computes the Maclaurin series to order 4. Entering

Series[Cos[x], {x, 0, 14}]

$$1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} - \frac{x^{10}}{3628800} + \frac{x^{12}}{479001600} - \frac{x^{14}}{81778291200} + O[x]^{15}$$

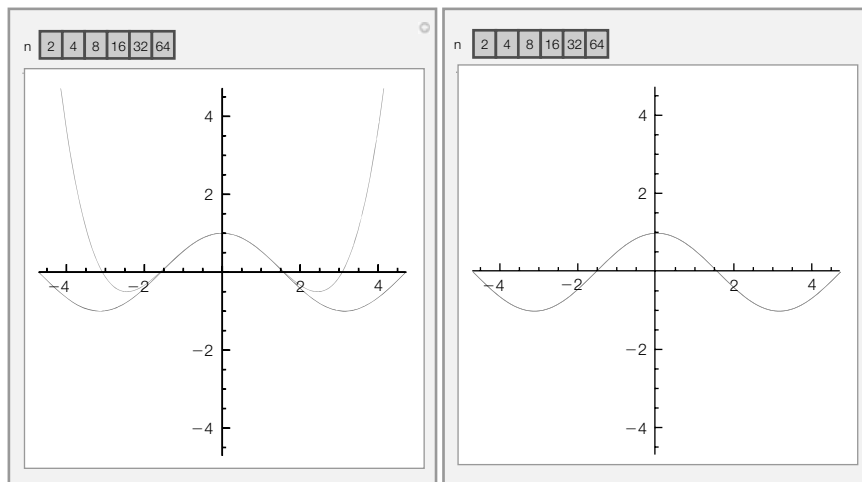
computes the Maclaurin series to order 14. In this case, the Maclaurin series for $\cos x$ converges to $\cos x$ for all real x . To graphically see this, we define the function `p`. Given n , `p[n]` returns the Maclaurin polynomial of degree n for $\cos x$.

p[n_] := Series[Cos[x], {x, 0, n}]/Normal

p[3]

$$1 - \frac{x^2}{2}$$

We then graph $\cos x$ together with the Maclaurin polynomial of degree $n = 2, 4, 8$, and 16 on the interval $[-3\pi/2, 3\pi/2]$ in Figure 3.54. Notice that as n increases, the graph of the Maclaurin polynomial more closely resembles the graph of $\cos x$.

**FIGURE 3.54**

Using `Manipulate` to investigate graphs of $y = \cos x$ together with plots of several of its Maclaurin polynomials

We would see the same pattern if we increased the length of the interval and the value of n .

```
Manipulate[Plot[Evaluate[Tooltip[{{Cos[x], p[n]}],  
{x, -3Pi/2, 3Pi/2}, PlotRange -> {{-3Pi/2, 3Pi/2}, {-3Pi/2, 3Pi/2}},  
AspectRatio -> Automatic,  
{n, 4}, {2, 4, 8, 16, 32, 64}, ControlType -> Setter]]
```

(b) After defining $f(x) = 1/x^2$, we compute the first 10 terms of the Taylor series for $f(x)$ about $x = 1$ with `Series`.

```
f[x_] = 1/x^2;  
p10 = Series[f[x], {x, 1, 10}]  
1 - 2(x - 1) + 3(x - 1)^2 - 4(x - 1)^3 + 5(x - 1)^4 - 6(x - 1)^5 + 7(x - 1)^6 - 8(x - 1)^7  
+ 9(x - 1)^8 - 10(x - 1)^9 + 11(x - 1)^10 + O[x - 1]^11  
1 - 2(x - 1) + 3(x - 1)^2 - 4(x - 1)^3 + 5(x - 1)^4 - 6(x - 1)^5 + 7(x - 1)^6 - 8(x - 1)^7  
+ 9(x - 1)^8 - 10(x - 1)^9 + 11(x - 1)^10 + O[x - 1]^11
```

In this case, the pattern for the series is relatively easy to see: The Taylor series for $f(x)$ about $x = 1$ is

$$\sum_{k=0}^{\infty} (-1)^k (k+1)(x-1)^k.$$

This series converges absolutely if

$$\lim_{k \rightarrow \infty} \left| \frac{(-1)^{k+1} (k+2)(x-1)^{k+1}}{(-1)^k (k+1)(x-1)^k} \right| = |x-1| < 1$$

or $0 < x < 2$. The series diverges if $x = 0$ and $x = 2$. In this case, the series converges to $f(x)$ on the interval $(0, 2)$.

```

a[x_, k_] = (-1)^k (k + 1) (x - 1)^k;
s1 = FullSimplify[Abs[a[x, k + 1]/a[x, k]]]
Abs[ $\frac{(2+k)(-1+x)}{1+k}$ ]

s2 = Limit[s1, k → Infinity]
Abs[-1 + x]

Reduce[s2 < 1, x]
0 < Re[x] < 2 && -√(2Re[x] - Re[x]^2) < Im[x] < √(2Re[x] - Re[x]^2)

```

To see this, we use **Manipulate** graph $f(x)$ together with the Taylor polynomial for $f(x)$ about $x = 1$ of degree n for large n . Regardless of the size of n , the graphs of $f(x)$ and the Taylor polynomial closely resemble each other on the interval $(0, 2)$ — but not at the endpoints or outside the interval. (See Figure 3.55.)

```

p[n_] := Series[f[x], {x, 1, n}]/Normal

Manipulate[Plot[Evaluate[Tooltip[{f[x], p[n]}]],
  {x, 0, 3}, PlotRange → {{0, 3}, {-3/2, 3/2}}, AspectRatio → Automatic],
  {{n, 10}, {5, 10, 15, 25, 30, 35, 40, 45, 50, 55, 60}}, ControlType → Setter]

```

3.4.6 Taylor's Theorem

Taylor's theorem states the relationship between $f(x)$ and the Taylor series for $f(x)$ about $x = x_0$.

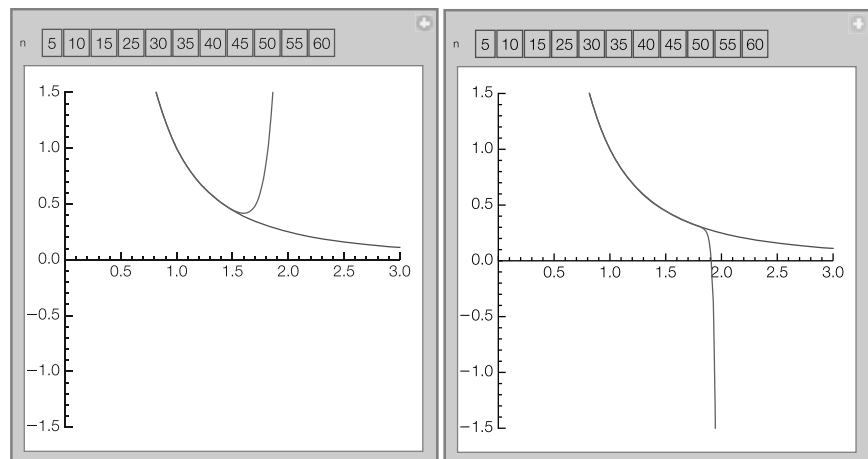


FIGURE 3.55

Graphs of $f(x)$ together with the various Taylor polynomials about $x = 1$

Theorem 16 (Taylor's Theorem). Let $y = f(x)$ have (at least) $n + 1$ derivatives on an interval I containing $x = x_0$. Then, for every number $x \in I$, there is a number z between x and x_0 so that

$$f(x) = p_n(x) + R_n(x),$$

where $p_n(x)$ is given by equation (3.28) and

$$R_n(x) = \frac{f^{(n+1)}(z)}{(n+1)!} (x - x_0)^{n+1}. \quad (3.29)$$

Example 3.4.8 Use Taylor's theorem to show that

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}$$

Solution Let $f(x) = \sin x$. Then, for each value of x , there is a number z between 0 and x so that $\sin x = p_n(x) + R_n(x)$, where $p_n(x) = \sum_{k=0}^n \frac{f^{(k)}(0)}{k!} x^k$ and $R_n(x) = \frac{f^{(n+1)}(z)}{(n+1)!} x^{n+1}$. Regardless of the value of n , $f^{(n+1)}(z)$ is one of $\sin z$, $-\sin z$, $\cos z$, or $-\cos z$, which are all bounded by 1. Then,

$$|\sin x - p_n(x)| = \left| \frac{f^{(n+1)}(z)}{(n+1)!} x^{n+1} \right|$$

$$|\sin x - p_n(x)| \leq \frac{1}{(n+1)!} |x|^{n+1}$$

and $x^n/n! \rightarrow 0$ as $n \rightarrow \infty$ for all real values of x .

You should remember that the number z in $R_n(x)$ is guaranteed to exist by Taylor's theorem. However, from a practical standpoint, you would rarely (if ever) need to compute the z value for a particular x value.

For illustrative purposes, we show the difficulties. Suppose we wish to approximate $\sin \pi/180$ using the Maclaurin polynomial of degree 4, $p_4(x) = x - \frac{1}{6}x^3$, for $\sin x$. The fourth remainder is $R_4(x) = \frac{1}{120} \cos z x^5$.

The Maclaurin polynomial of degree 4 for $\sin x$ is

$$\sum_{k=0}^4 \frac{f^{(k)}(0)}{k!} x^k = 0 + x + 0 \cdot x^2 + \frac{-1}{3!} x^3 + 0 \cdot x^4.$$

```
Clear[f]
f[x_] = Sin[x];
r5 = D[f[z], {z, 5}]/5! x^5
1/120 x^5 Cos[z]
```

If $x = \pi/180$, there is a number z between 0 and $\pi/180$ so that

$$\left| R_4\left(\frac{\pi}{180}\right) \right| = \frac{1}{120} \cos z \left(\frac{\pi}{180}\right)^5$$

$$\leq \frac{1}{120} \left(\frac{\pi}{180}\right)^5 \approx 0.135 \times 10^{-10},$$

which shows us that the maximum the error can be is $\frac{1}{120} \left(\frac{\pi}{180}\right)^5 \approx 0.135 \times 10^{-10}$.

```
maxerror = N[1/120*(Pi/180)^5]
1.349601623163255*^-11
```

Abstractly, the exact error can be computed. By Taylor's theorem, z satisfies

$$f\left(\frac{\pi}{180}\right) = p_4\left(\frac{\pi}{180}\right) + R_4\left(\frac{\pi}{180}\right)$$

$$\sin \frac{\pi}{180} = \frac{1}{180}\pi - \frac{1}{34992000}\pi^3 + \frac{1}{22674816000000}\pi^5 \cos z$$

$$0 = \frac{1}{180}\pi - \frac{1}{34992000}\pi^3 + \frac{1}{22674816000000}\pi^5 \cos z - \sin \frac{\pi}{180}.$$

We graph the right-hand side of this equation with **Plot** in Figure 3.56. The exact value of z is the z -coordinate of the point where the graph intersects the z -axis.

```
p4 = Series[f[x], {x, 0, 4}]/Normal
```

```
x - x^3/6
```

```
exval = Sin[Pi/180]
```

```
p4b = p4/.x -> Pi/180
```

```
r5b = r5/.x -> Pi/180
```

```
Sin [Pi/180]
```

```
pi/180 - pi^3/34992000
```

```
pi^5 Cos[z]/22674816000000
```

```
toplot = r5b + p4b - exval;
```

```
Plot[toplot, {z, 0, Pi/180}]
```

We can use **FindRoot** to approximate z , if we increase the number of digits carried in floating point calculations with **WorkingPrecision**.

```
exz = FindRoot[toplot==0, {z, 0, .004}, WorkingPrecision -> 32]
{z -> 0.0038086149165541606417429516417308}
```

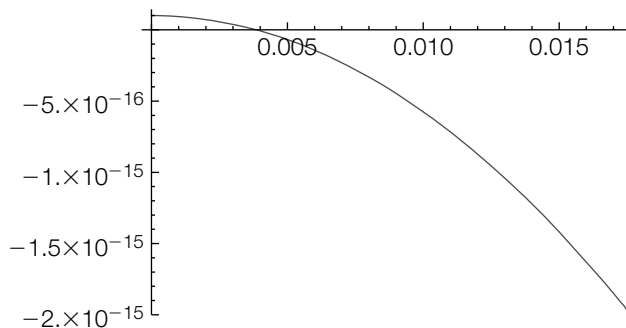


FIGURE 3.56

Finding z

Alternatively, we can compute the exact value of z with `Solve`

Because Mathematica uses inverse functions in this calculation, it issues several warning messages that we have omitted for length considerations.

$$\mathbf{cz} = \text{Solve}[\text{tplot} == 0, z] \\ \left\{ \left\{ z \rightarrow \text{ArcCos} \left[\frac{648000(-194400\pi + \pi^3 + 34992000 \sin[\frac{\pi}{180}])}{\pi^5} \right] \right\}, \right. \\ \left. \left\{ z \rightarrow \text{ArcCos} \left[\frac{648000(-194400\pi + \pi^3 + 34992000 \sin[\frac{\pi}{180}])}{\pi^5} \right] \right\} \right\}$$

and then approximate the result with `N`.

$$\mathbf{N[cz]} \\ \{\{z \rightarrow -0.00384232\}, \{z \rightarrow 0.00384232\}\}$$

3.4.7 Other Series

In calculus, we learn that the power series $f(x) = \sum_{k=0}^{\infty} a_k (x - x_0)^k$ is differentiable and integrable on its interval of convergence. However, for series, that are not power series, this result is not generally true. For example, in more advanced courses, we learn that the function

$$f(x) = \sum_{k=0}^{\infty} \frac{1}{2^k} \sin(3^k x)$$

is continuous for all values of x but nowhere differentiable. We can use Mathematica to help us see why this function is not differentiable. Let

$$f_n(x) = \sum_{k=0}^n \frac{1}{2^k} \sin(3^k x).$$

Notice that $f_n(x)$ is defined recursively by $f_0(x) = \sin x$ and $f_n(x) = f_{n-1}(x) + \frac{1}{2^n} \sin(3^n x)$. We use Mathematica to recursively define $f_n(x)$.

$$\begin{aligned} \mathbf{Clear[f]} \\ \mathbf{f[0] = Sin[x];} \\ \mathbf{f[k_]:=f[k] = f[k-1] + \frac{\text{Sin}[3^k x]}{2^k}} \end{aligned}$$

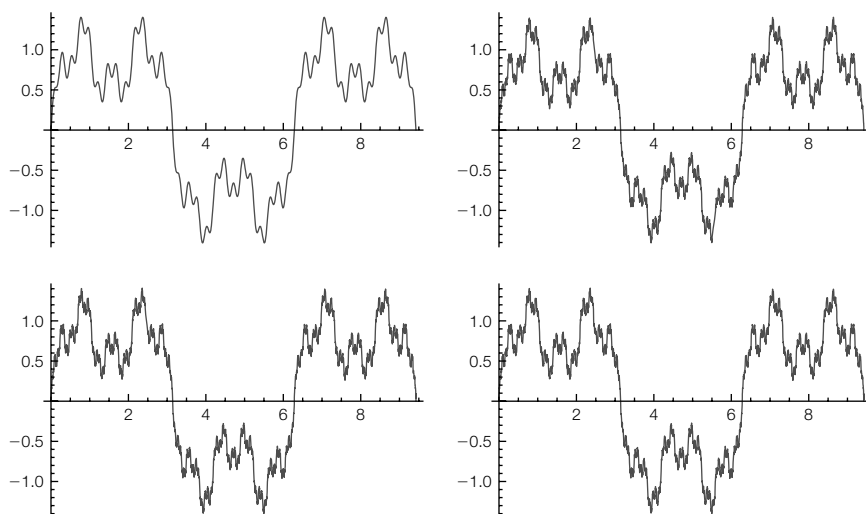
We define $f_n(x)$ using the form

$$\mathbf{f[n_]:=f[n] = ...}$$

so that Mathematica “remembers” the values it computes. Thus, to compute `f[5]`, Mathematica uses the previously computed values, namely `f[4]`, to compute `f[5]`. Note that we can produce the same results by defining $f_n(x)$ with the command

$$\mathbf{f[n_]:=...}$$

However, the disadvantage of defining $f_n(x)$ in this manner is that Mathematica does not “remember” the previously computed values and thus takes longer to compute $f_n(x)$ for larger values of n .

**FIGURE 3.57**

Approximating a function that is continuous everywhere but nowhere differentiable

Next, we use Table to generate $f_3(x)$, $f_6(x)$, $f_9(x)$, and $f_{12}(x)$.

```
tograph = Table[f[n], {n, 3, 12, 3}];
```

We now graph each of these functions and show the results as a graphics array with GraphicsGrid in Figure 3.57. (Note that you do not need to include the option DisplayFunction->Identity to suppress the resulting output unless you forget to include the semicolon at the end of the command.)

```
graphs = Table[Plot[Evaluate[tograph[[i]]], {x, 0, 3π},
  DisplayFunction -> Identity], {i, 1, 4}];
toshow = Partition[graphs, 2];
Show[GraphicsGrid[toshow]]
```

From these graphs, we see that for large values of n , the graph of $f_n(x)$, although actually smooth, appears “jagged” and thus we might suspect that $f(x) = \lim_{n \rightarrow \infty} f_n(x) = \sum_{k=0}^{\infty} \frac{1}{2^k} \sin(3^k x)$ is indeed continuous everywhere but nowhere differentiable.

3.5 MULTIVARIABLE CALCULUS

Mathematica is useful in investigating functions involving more than one variable. In particular, the graphical analysis of functions that depend on two (or more) variables is enhanced with the help of Mathematica’s graphics capabilities.

3.5.1 Limits of Functions of Two Variables

Mathematica's graphics and numerical capabilities are helpful in investigating limits of functions of two variables.

Example 3.5.1 Show that the limit $\lim_{(x,y) \rightarrow (0,0)} \frac{x^2 - y^2}{x^2 + y^2}$ does not exist.

Solution We begin by defining $f(x,y) = (x^2 - y^2)/(x^2 + y^2)$. Next, we use `Plot3D` to graph $z = f(x,y)$ for $-1/2 \leq x \leq 1/2$ and $-1/2 \leq y \leq 1/2$. `ContourPlot` is used to graph several level curves on the same rectangle. (See Figure 3.58.) (To define a function of two variables, $f(x,y) = \text{expression in } x \text{ and } y$, enter `f[x_,y_] = expression in x and y`. `Plot3D[f[x,y], {a,x,b},{y,c,d}]` generates a basic graph of $z = f(x,y)$ for $a \leq x \leq b$ and $c \leq y \leq d$)

```
Clear[f]
f[x_,y_] = (x^2 - y^2)/(x^2 + y^2);
p1 = Plot3D[f[x,y], {x, -2, 2}, {y, -2, 2}, PlotPoints -> 40];
c1 = ContourPlot[f[x,y], {x, -2, 2}, {y, -2, 2}, ContourShading -> False,
  Axes -> Automatic, AxesOrigin -> {0,0}, PlotPoints -> 60, Contours -> 20];
Show[GraphicsRow[{p1,c1}]]
```

When you slide the cursor over the contours in the contour plot, the contour values are displayed.

From the graph of the level curves, we suspect that the limit does not exist because we see that near $(0,0)$, $z = f(x,y)$ attains many different values. We obtain further evidence that the limit does not exist by computing the value of $z = f(x,y)$ for various points chosen randomly near $(0,0)$. We use `Table` and `RandomReal` to generate 10 ordered pairs (x,y) for x and y "close to" 0. Because `RandomReal` is included in the calculation, your results will almost certainly be different from those here.

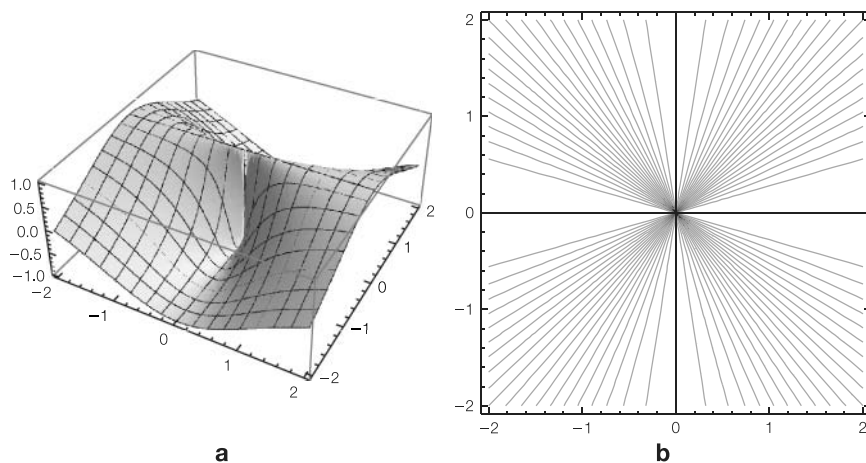


FIGURE 3.58

(a) Three-dimensional and (b) contour plots of $f(x,y)$

```
pts = Table[RandomReal[{-10-1, 10-1}], {i, 1, 10}, {2}]
{{0.08560753251471961, -0.06804168963904592},
{0.004371826092799417, 0.005941850437676466},
{-0.0009189852964073855, 0.00090260881737971},
{-0.00006532557511289984, 0.00006629693177194204},
{-9.419651789936537*-6, -1.2215350865693182*-6},
{-5.743976531479158*-7, 5.766874942905546*-7},
{-7.626065924242957*-8, 5.4979380353427926*-8},
{-8.100264683497016*-10, -3.253471996451157*-9},
{5.270631463415014*-12, 7.862252513620563*-10},
{-3.633733884546907*-11, -3.351445796649152*-11}}
```

Next, we define a function g that given an ordered pair (x, y) ($\{x, y\}$ in Mathematica), $g((x, y))$ returns the ordered triple $(x, y, f(x, y))$ ($\{x, y, f[x, y]\}$ in Mathematica).

```
g[{x_, y_}] = {x, y, f[x, y]}
{ x, y,  $\frac{x^2 - y^2}{x^2 + y^2}$  }
```

We then use `Map` to apply g to the list `pts`.

```
Map[g, pts]//TableForm
0.0856075          -0.0680417          0.225699
0.00437183         0.00594185          -0.297559
-0.000918985       0.000902609         0.0179789
-0.0000653256      0.0000662969        -0.0147589
-9.419651789936537'-6 -1.2215350865693182'-6 0.966923
-5.743976531479158'-7 5.766874942905546'-7 -0.00397856
-7.626065924242957'-8 5.4979380353427926'-8 0.316002
-8.100264683497016'-10 -3.253471996451157'-9 -0.883261
5.270631463415014'-12 7.862252513620563'-10 -0.99991
-3.633733884546907'-11 -3.351445796649152'-11 0.0806931
```

From the third column, we see that $z = f(x, y)$ does not appear to approach any particular value for points chosen randomly near $(0, 0)$. In fact, along the line $y = mx$ we see that $f(x, y) = f(x, mx) = \frac{(1 - m^2)}{(1 + m^2)}$. Hence, as $(x, y) \rightarrow (0, 0)$ along $y = mx$, $f(x, y) = f(x, mx) \rightarrow \frac{1 - m^2}{1 + m^2}$. Thus, $f(x, y)$ does not have a limit as $(x, y) \rightarrow (0, 0)$.

We choose lines of the form $y = mx$ because near $(0, 0)$ the level curves of $z = f(x, y)$ look like lines of the form $y = mx$.

```
v1 = Simplify[f[x, mx]]
```

```
 $\frac{1 - m^2}{1 + m^2}$ 
```

```
v1/.m → 0
```

```
v1/.m → 1
```

```
v1/.m → 1/2
```

```
1
```

```
0
```

```
 $\frac{3}{5}$ 
```

In some cases, you can establish that a limit does not exist by converting to polar coordinates. For example, in polar coordinates, $f(x, y) = \frac{x^2 - y^2}{x^2 + y^2}$ becomes $f(r \cos \theta, r \sin \theta) = 2 \cos^2 \theta - 1$

Simplify[f[rCos[t], rSin[t]]]

Cos[2t]

and

$$\lim_{(x,y) \rightarrow (0,0)} f(x, y) = \lim_{r \rightarrow 0} f(r \cos \theta, r \sin \theta) = \lim_{r \rightarrow 0} 2 \cos^2 \theta - 1 = 2 \cos^2 \theta - 1 = \cos 2\theta$$

depends on θ .

3.5.2 Partial and Directional Derivatives

Partial derivatives of functions of two or more variables are computed with Mathematica using **D**. For $z = f(x, y)$,

1. **D[f[x,y],x]** computes $\frac{\partial f}{\partial x} = f_x(x, y)$,
2. **D[f[x,y],y]** computes $\frac{\partial f}{\partial y} = f_y(x, y)$,
3. **D[f[x,y],{x,n}]** computes $\frac{\partial^n f}{\partial x^n}$,
4. **D[f[x,y],y,x]** computes $\frac{\partial^2 f}{\partial y \partial x} = f_{xy}(x, y)$, and
5. **D[f[x,y],{x,n},{y,m}]** computes $\frac{\partial^{n+m} f}{\partial^n x \partial^m y}$.
6. You can use the  button located on the **BasicMathInput** palette to create templates to compute partial derivatives.

The calculations are carried out similarly for functions of more than two variables.

Example 3.5.2 Calculate $f_x(x, y)$, $f_y(x, y)$, $f_{xy}(x, y)$, $f_{yx}(x, y)$, $f_{xx}(x, y)$, and $f_{yy}(x, y)$ if $f(x, y) = \sin \sqrt{x^2 + y^2 + 1}$.

Solution After defining $f(x, y) = \sin \sqrt{x^2 + y^2 + 1}$,

f[x_, y_] = Sin[Sqrt[x^2 + y^2 + 1]];

we illustrate the use of **D** to compute the partial derivatives. Entering

D[f[x, y], x]

$$\frac{x \cos \left[\sqrt{1 + x^2 + y^2} \right]}{\sqrt{1 + x^2 + y^2}}$$

computes $f_x(x, y)$. Entering

D[f[x, y], y]

$$\frac{y \cos \left[\sqrt{1+x^2+y^2} \right]}{\sqrt{1+x^2+y^2}}$$

computes $f_y(x, y)$. Entering

D[f[x, y], x, y]//Together

$$\frac{-xy \cos \left[\sqrt{1+x^2+y^2} \right] - xy \sqrt{1+x^2+y^2} \sin \left[\sqrt{1+x^2+y^2} \right]}{(1+x^2+y^2)^{3/2}}$$

computes $f_{yx}(x, y)$. Entering

D[f[x, y], y, x]//Together

$$\frac{-xy \cos \left[\sqrt{1+x^2+y^2} \right] - xy \sqrt{1+x^2+y^2} \sin \left[\sqrt{1+x^2+y^2} \right]}{(1+x^2+y^2)^{3/2}}$$

computes $f_{xy}(x, y)$. Remember that under appropriate assumptions, $f_{xy}(x, y) = f_{yx}(x, y)$. Entering

D[f[x, y], {x, 2}]/Together

$$\frac{\cos \left[\sqrt{1+x^2+y^2} \right] + y^2 \cos \left[\sqrt{1+x^2+y^2} \right] - x^2 \sqrt{1+x^2+y^2} \sin \left[\sqrt{1+x^2+y^2} \right]}{(1+x^2+y^2)^{3/2}}$$

computes $f_{xx}(x, y)$. Entering

D[f[x, y], {y, 2}]/Together

$$\frac{\cos \left[\sqrt{1+x^2+y^2} \right] + x^2 \cos \left[\sqrt{1+x^2+y^2} \right] - y^2 \sqrt{1+x^2+y^2} \sin \left[\sqrt{1+x^2+y^2} \right]}{(1+x^2+y^2)^{3/2}}$$

computes $f_{yy}(x, y)$.

The **directional derivative** of $z = f(x, y)$ in the direction of the unit vector $\mathbf{u} = \cos \theta \mathbf{i} + \sin \theta \mathbf{j}$ is

$$D_{\mathbf{u}} f(x, y) = f_x(x, y) \cos \theta + f_y(x, y) \sin \theta,$$

provided that $f_x(x, y)$ and $f_y(x, y)$ both exist.

If $f_x(x, y)$ and $f_y(x, y)$ both exist, the **gradient** of $f(x, y)$ is the vector-valued function

$$\nabla f(x, y) = f_x(x, y) \mathbf{i} + f_y(x, y) \mathbf{j} = \langle f_x(x, y), f_y(x, y) \rangle.$$

Notice that if $\mathbf{u} = \langle \cos \theta, \sin \theta \rangle$,

$$D_{\mathbf{u}} f(x, y) = \nabla f(x, y) \cdot \langle \cos \theta, \sin \theta \rangle.$$

The vectors $\mathbf{i}$ and $\mathbf{j}$ are defined by $\mathbf{i} = \langle 1, 0 \rangle$ and $\mathbf{j} = \langle 0, 1 \rangle$.

Calculus of vector-valued functions is discussed in more detail in Chapter 5.

Example 3.5.3 Let $f(x, y) = 6x^2y - 3x^4 - 2y^3$. (a) Find $D_{\mathbf{u}} f(x, y)$ in the direction of $\mathbf{v} = \langle 3, 4 \rangle$. (b) Compute $D_{(3/5, 4/5)} f \left(\frac{1}{3} \sqrt{9 + 3\sqrt{3}}, 1 \right)$. (c) Find an equation of the line tangent to the graph of $6x^2y - 3x^4 - 2y^3 = 0$ at the point $\left(\frac{1}{3} \sqrt{9 + 3\sqrt{3}}, 1 \right)$.

Solution After defining $f(x, y) = 6x^2y - 3x^4 - 2y^3$, we graph $z = f(x, y)$ with Plot3D in Figure 3.59, illustrating the PlotPoints, PlotRange, and ViewPoint options.

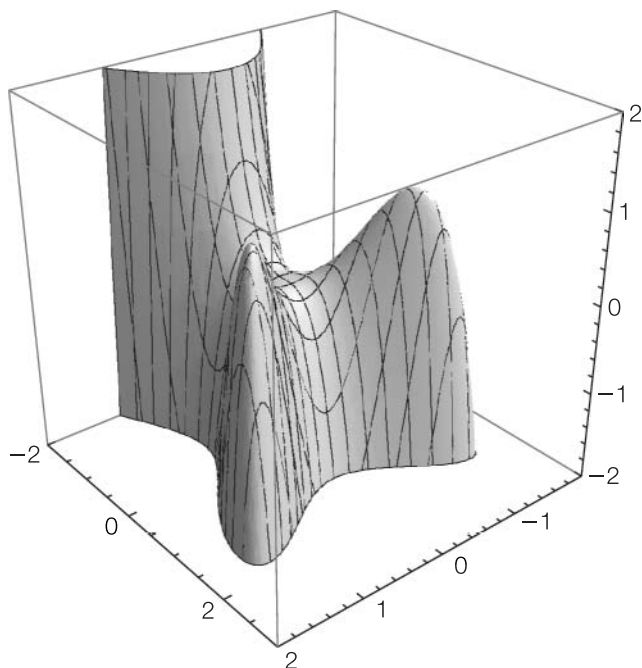


FIGURE 3.59

$f(x, y) = 6x^2y - 3x^4 - 2y^3$ for $-2 \leq x \leq 2$ and $-2 \leq y \leq 3$

```
f[x_, y_] = 6x^2y - 3x^4 - 2y^3;  
Plot3D[f[x, y], {x, -2, 2}, {y, -2, 3}, PlotPoints -> 50,  
PlotRange -> {{-2, 2}, {-2, 3}, {-2, 2}},  
BoxRatios -> {1, 1, 1}, ViewPoint -> {1.887, 2.309, 1.6},  
ClippingStyle -> None]
```

(a) A unit vector, $\mathbf{u}$, in the same direction as $\mathbf{v}$ is

$$\mathbf{u} = \left\langle \frac{3}{\sqrt{3^2 + 4^2}}, \frac{4}{\sqrt{3^2 + 4^2}} \right\rangle = \left\langle \frac{3}{5}, \frac{4}{5} \right\rangle.$$

v = {3, 4};

u = v/Sqrt[v . v]

{3/5, 4/5}

Then, $D_{\mathbf{u}}f(x, y) = \langle f_x(x, y), f_y(x, y) \rangle \cdot \mathbf{u}$, calculated in du.

gradf = {D[f[x, y], x], D[f[x, y], y]}

{-12x^3 + 12xy, 6x^2 - 6y^2}

du = Simplify[gradf . u]

-12/5 (-2x^2 + 3x^3 - 3xy + 2y^2)

(b) $D_{(3/5, 4/5)} f\left(\frac{1}{3}\sqrt{9+3\sqrt{3}}, 1\right)$ is calculated by evaluating du if $x = \frac{1}{3}\sqrt{9+3\sqrt{3}}$ and $y = 1$.

$$\begin{aligned} du &= du/.{x \rightarrow 1/3\text{Sqrt}[9 + 3\text{Sqrt}[3]], y \rightarrow 1}/\text{Simplify} \\ &\frac{4}{5} \left(2\sqrt{3} - 3\sqrt{3 + \sqrt{3}} \right) \end{aligned}$$

(c) The gradient is evaluated if $x = \frac{1}{3}\sqrt{9+3\sqrt{3}}$ and $y = 1$.

$$\begin{aligned} nvec &= \text{gradf}/.{x \rightarrow 1/3\text{Sqrt}[9 + 3\text{Sqrt}[3]], y \rightarrow 1}/\text{Simplify} \\ &\left\{ -4\sqrt{3 + \sqrt{3}}, 2\sqrt{3} \right\} \end{aligned}$$

Generally, $\nabla f(x, y)$ is perpendicular to the level curves of $z = f(x, y)$, so

$$nvec = \nabla f\left(\frac{1}{3}\sqrt{9+3\sqrt{3}}, 1\right) = \left\langle f_x\left(\frac{1}{3}\sqrt{9+3\sqrt{3}}, 1\right), f_y\left(\frac{1}{3}\sqrt{9+3\sqrt{3}}, 1\right) \right\rangle$$

An equation of the line L containing (x_0, y_0) and perpendicular to $\mathbf{n} = \langle a, b \rangle$ is $a(x - x_0) + b(y - y_0) = 0$.

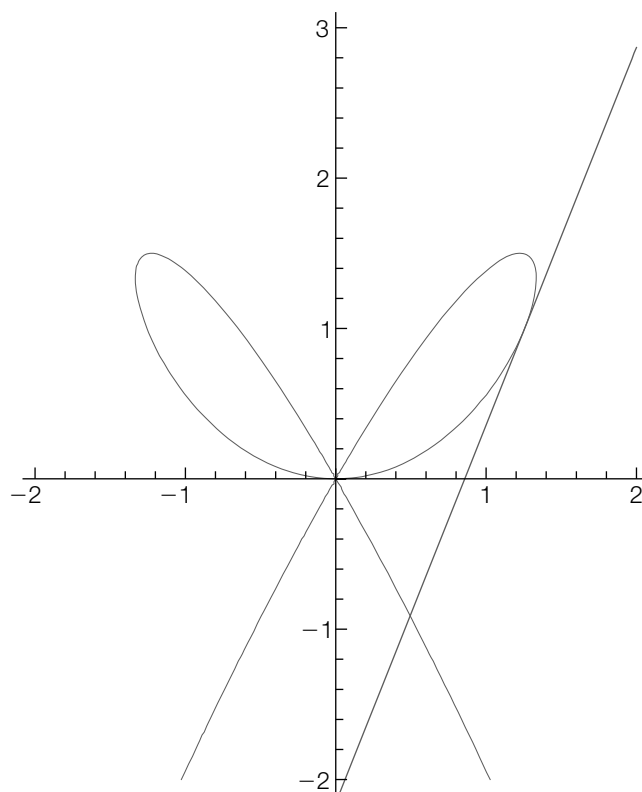
is perpendicular to $f(x, y) = 0$ at the point $\left(\frac{1}{3}\sqrt{9+3\sqrt{3}}, 1\right)$. Thus, an equation of the line tangent to the graph of $f(x, y) = 0$ at the point $\left(\frac{1}{3}\sqrt{9+3\sqrt{3}}, 1\right)$ is

$$f_x\left(\frac{1}{3}\sqrt{9+3\sqrt{3}}, 1\right)\left(x - \frac{1}{3}\sqrt{9+3\sqrt{3}}\right) + f_y\left(\frac{1}{3}\sqrt{9+3\sqrt{3}}, 1\right)(y - 1) = 0,$$

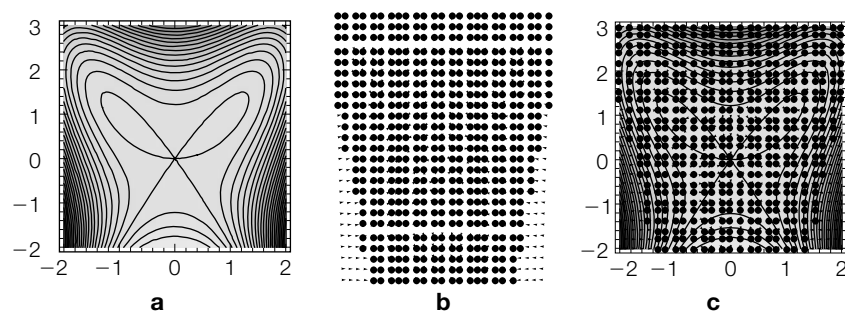
which we solve for y with `Solve`. We confirm this result by graphing $f(x, y) = 0$ using `ContourPlot` in `conf` and then graphing the tangent line in `tanplot`. `tanplot` and `conf` are shown together with `Show` in Figure 3.60.

```
conf = ContourPlot[f[x, y]==0, {x, -2, 2}, {y, -2, 2}, PlotPoints -> 60,
  ContourShading -> False, Frame -> False, Axes -> Automatic,
  AxesOrigin -> {0, 0}];
tanline = Solve[nvec[[1]](x - 1/3Sqrt[9 + 3Sqrt[3]]) + nvec[[2]](y - 1)==0,
  y]
{ { y -> - (2 + Sqrt[3] - 2 Sqrt[3 + Sqrt[3] x]) / Sqrt[3] } }
Evaluate[y[x]/.tanline[[1]]]
{ - (2 + Sqrt[3] - 2 Sqrt[3 + Sqrt[3] x]) / Sqrt[3] } [x]
tanplot = Plot[Evaluate[y/.tanline], {x, -2, 2}];
Show[conf, tanplot, PlotRange -> {{-2, 2}, {-2, 3}}, AspectRatio ->
Automatic]
```

More generally, we use `ContourPlot` together with the `PlotGradientField` function, which is contained in the `VectorFieldPlots` package, to illustrate that the gradient vectors are perpendicular to the level curves of $z = f(x, y)$ in Figure 3.61.

**FIGURE 3.60**

Level curve of $f(x, y)$ together with a tangent line

**FIGURE 3.61**

(a) Level curves of $z = f(x, y)$. (b) Gradient field of $z = f(x, y)$. (c) The gradient together with several level curves

```

<< VectorFieldPlots";
p1 = ContourPlot[f[x, y], {x, -2, 2}, {y, -2, 3},
  Contours -> 25, ContourStyle -> Black]
p2 = PlotGradientField[f[x, y], {x, -2, 2}, {y, -2, 3}, PlotPoints -> 25]
Show[p1, p2]
Show[GraphicsRow[{p1, p2, Show[{p1, p2}]}]]

```

Example 3.5.4 Let

$$f(x, y) = (y - 1)^2 e^{-(x+1)^2 - y^2} - \frac{10}{3} \left(-x^5 + \frac{1}{5}y - y^3 \right) e^{-x^2 - y^2} - \frac{1}{9} e^{-x^2 - (y+1)^2}.$$

Calculate $\nabla f(x, y)$ and then graph $\nabla f(x, y)$ together with several level curves of $f(x, y)$.

Solution We begin by defining and graphing $z = f(x, y)$ with Plot3D in Figure 3.62(a).

```

Clear[f]
f[x_, y_] = (y - 1)^2 Exp[-(x + 1)^2 - y^2] -
  10/3(-x^5 + 1/5y - y^3)Exp[-x^2 - y^2] -
  1/9Exp[-x^2 - (y + 1)^2];
p1 = Plot3D[f[x, y], {x, -3, 3}, {y, -3, 3},
  ViewPoint -> {-1.99, 2.033, 1.833},
  PlotRange -> All];
conf = ContourPlot[f[x, y], {x, -3, 3}, {y, -3, 3},
  PlotPoints -> 60, Contours -> 30, ContourShading -> False,
  Frame -> False, Axes -> Automatic,
  AxesOrigin -> {0, 0}];

```

In the three-dimensional plot, notice that z appears to have six relative extrema: three relative maxima and three relative minima. We also graph several level curves of $f(x, y)$ with ContourPlot and name the resulting graphic conf.

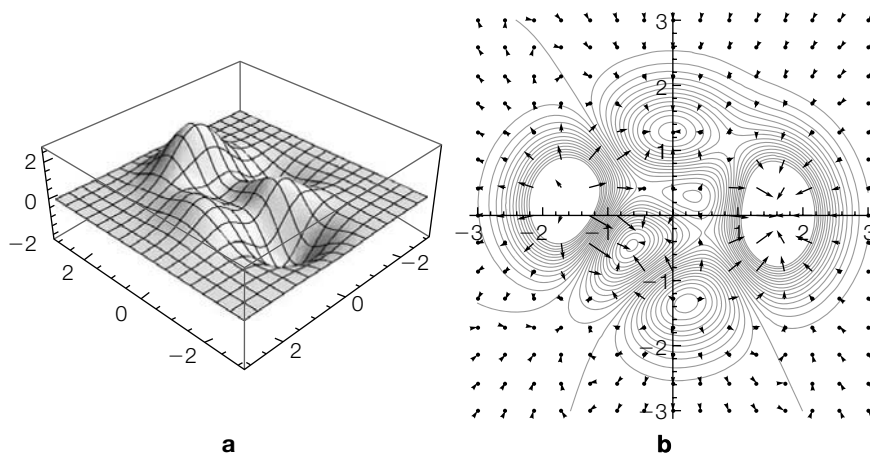
Next we calculate $f_x(x, y)$ and $f_y(x, y)$ using Simplify and D. The gradient is the vector-valued function $\langle f_x(x, y), f_y(x, y) \rangle$.

```

gradf = {D[f[x, y], x], D[f[x, y], y]}/Simplify
{
  -2/9 e^{-2x - x^2 - (1+y)^2}
  (-e^{2x} x + 9e^{2y} (1 + x)(-1 + y)^2 + 3e^{1+2x+2y} x (-25x^3 + 10x^5 - 2y + 10y^3)) ,
  -2/9 e^{-2x - x^2 - (1+y)^2}
  (-e^{2x} (1 + y) + 9e^{2y} (1 - 2y^2 + y^3) + e^{1+2x+2y} (3 + 30x^5 y - 51y^2 + 30y^4)) }

```

To graph the gradient, we use PlotGradientField, which is contained in the VectorFieldPlots package. We use PlotGradientField to graph the gradient, naming the resulting graphic gradfplot. gradfplot and conf are displayed together using Show in Figure 3.62(b).

**FIGURE 3.62**

(a) $f(x, y)$ for $-3 \leq x \leq 3$ and $-3 \leq y \leq 2$. (b) Contour plot of $f(x, y)$ along with several gradient vectors

```
<< VectorFieldPlots`;  
gradfplot = PlotGradientField[f[x, y], {x, -3, 3},  
    {y, -3, 3}];  
Show[GraphicsRow[{p1, Show[conf, gradfplot]}]]
```

In the result (see Figure 3.62(b)), notice that the gradient is perpendicular to the level curves; the gradient is pointing in the direction of maximal increase of $z = f(x, y)$.

Classifying Critical Points

Let $z = f(x, y)$ be a real-valued function of two variables with continuous second-order partial derivatives. A **critical point** of $z = f(x, y)$ is a point (x_0, y_0) in the interior of the domain of $z = f(x, y)$ for which

$$f_x(x_0, y_0) = 0 \quad \text{and} \quad f_y(x_0, y_0) = 0.$$

Critical points are classified by the *second derivatives (or partials) test*.

Theorem 17 (Second Derivatives Test). Let (x_0, y_0) be a critical point of a function $z = f(x, y)$ of two variables and let

$$d = f_{xx}(x_0, y_0) f_{yy}(x_0, y_0) - [f_{xy}(x_0, y_0)]^2. \quad (3.30)$$

1. If $d > 0$ and $f_{xx}(x_0, y_0) > 0$, then $z = f(x, y)$ has a **relative (or local) minimum** at (x_0, y_0) .
2. If $d > 0$ and $f_{xx}(x_0, y_0) < 0$, then $z = f(x, y)$ has a **relative (or local) maximum** at (x_0, y_0) .

3. If $d < 0$, then $z = f(x, y)$ has a **saddle point** at (x_0, y_0) .
4. If $d = 0$, no conclusion can be drawn and (x_0, y_0) is called a **degenerate critical point**.

Example 3.5.5 Find the relative maximum, relative minimum, and saddle points of $f(x, y) = -2x^2 + x^4 + 3y - y^3$.

Solution After defining $f(x, y)$, the critical points are found with `Solve` and named `critpts`.

```
f[x_, y_] = -2x^2 + x^4 + 3y - y^3;
critpts = Solve[{D[f[x, y], x]==0, D[f[x, y], y]==0}, {x, y}]
{{x -> -1, y -> -1}, {x -> -1, y -> 1}, {x -> 0, y -> -1},
{x -> 0, y -> 1}, {x -> 1, y -> -1}, {x -> 1, y -> 1}}
```

We then define `dfxx`. Given (x_0, y_0) , `dfxx` (x_0, y_0) returns the ordered quadruple x_0, y_0 , equation (3.30) evaluated at (x_0, y_0) , and $f_{xx}(x_0, y_0)$.

```
dfxx[x0_, y0_] = {x0, y0,
D[f[x, y], {x, 2}]D[f[x, y], {y, 2}] - D[f[x, y], x, y]^2/,
{x -> x0, y -> y0}, D[f[x, y], {x, 2}]/.{x -> x0, y -> y0}}
{x0, y0, -6(-4 + 12x0^2)y0, -4 + 12x0^2}
```

For example,

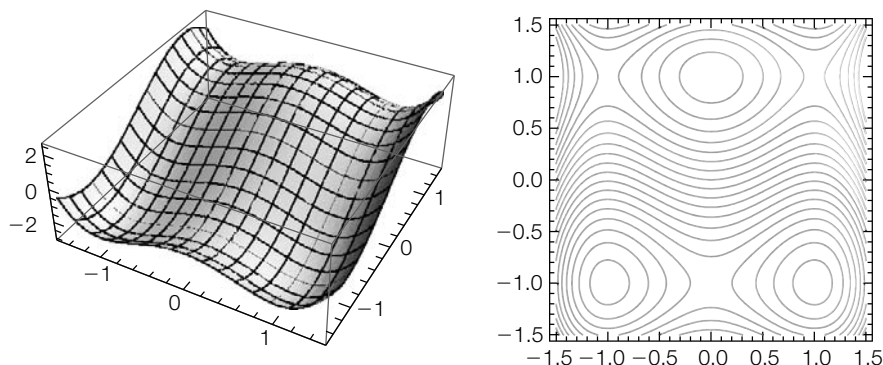
```
dfxx[0, 1]
{0, 1, 24, -4}
```

shows us that a relative maximum occurs at $(0, 1)$. We then use `/.` (`ReplaceAll`) to substitute the values in each element of `critpts` into `dfxx`.

```
dfxx[x, y]/.critpts//TableForm
-1 -1 48 8
-1 1 -48 8
0 -1 -24 -4
0 1 24 -4
1 -1 48 8
1 1 -48 8
```

From the result, we see that $(0, 1)$ results in a relative maximum, $(0, -1)$ results in a saddle, $(1, 1)$ results in a saddle, $(1, -1)$ results in a relative minimum, $(-1, 1)$ results in a saddle, and $(-1, -1)$ results in a relative minimum. We confirm these results graphically with a three-dimensional plot generated with `Plot3D` and a contour plot generated with `ContourPlot` in Figure 3.63.

```
p1 = Plot3D[f[x, y], {x, -3/2, 3/2}, {y, -3/2, 3/2}, PlotPoints -> 40];
p2 = ContourPlot[f[x, y], {x, -3/2, 3/2}, {y, -3/2, 3/2},
PlotPoints -> 40, ContourShading -> False, Contours -> 20];
Show[GraphicsRow[{p1, p2}]]
```

**FIGURE 3.63**

(a) Three-dimensional and (b) contour plots of $f(x, y)$

In the contour plot, notice that near relative extrema, the level curves look like circles, whereas near saddles they look like hyperbolas.

If the second derivatives test fails, graphical analysis is especially useful.

Example 3.5.6 Find the relative maximum, relative minimum, and saddle points of $f(x, y) = x^2 + x^2y^2 + y^4$.

Solution Initially we proceed in the exact same manner as in the previous example: We define $f(x, y)$ and compute the critical points. Several complex solutions are returned, which we ignore.

```
f[x_, y_] = x^2 + x^2y^2 + y^4;
critpts = Solve[{D[f[x, y], x] == 0, D[f[x, y], y] == 0}, {x, y}]
{{x -> 0, y -> 0}, {x -> -sqrt(2), y -> -i}, {x -> -sqrt(2), y -> i},
{x -> sqrt(2), y -> -i}, {x -> sqrt(2), y -> i}, {y -> 0, x -> 0}, {y -> 0, x -> 0}}
```

We then compute the value of (3.30) at the real critical point, and the value of $f_{xx}(x, y)$ at this critical point.

```
dfxx[x0_, y0_] = {x0, y0,
D[f[x, y], {x, 2}]D[f[x, y], {y, 2}] - D[f[x, y], x, y]^2 /
{x -> x0, y -> y0}, D[f[x, y], {x, 2}] /. {x -> x0, y -> y0}}
{x0, y0, -16x0^2y0^2 + (2 + 2y0^2) (2x0^2 + 12y0^2), 2 + 2y0^2}
dfxx[0, 0]
{0, 0, 0, 2}
```

The result shows us that the second derivatives test fails at $(0, 0)$.

```

p1 = Plot3D[f[x, y], {x, -1, 1}, {y, -1, 1}, BoxRatios -> Automatic];
p2 = ContourPlot[f[x, y], {x, -1, 1}, {y, -1, 1}, PlotPoints -> 40,
Contours -> 20, ContourShading -> False];
Show[GraphicsRow[{p1, p2}]]

```

However, the contour plot of $f(x, y)$ near $(0, 0)$ indicates that an extreme value occurs at $(0, 0)$. The three-dimensional plot shows that $(0, 0)$ is a relative minimum. (See Figure 3.64.)

Tangent Planes

Let $z = f(x, y)$ be a real-valued function of two variables. If both $f_x(x_0, y_0)$ and $f_y(x_0, y_0)$ exist, then an equation of the plane tangent to the graph of $z = f(x, y)$ at the point $(x_0, y_0, f(x_0, y_0))$ is given by

$$f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0) - (z - z_0) = 0, \quad (3.31)$$

where $z_0 = f(x_0, y_0)$. Solving for z yields the function (of two variables)

$$z = f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0) + z_0. \quad (3.32)$$

Symmetric equations of the line perpendicular to the surface $z = f(x, y)$ at the point (x_0, y_0, z_0) are given by

$$\frac{x - x_0}{f_x(x_0, y_0)} = \frac{y - y_0}{f_y(x_0, y_0)} = \frac{z - z_0}{-1} \quad (3.33)$$

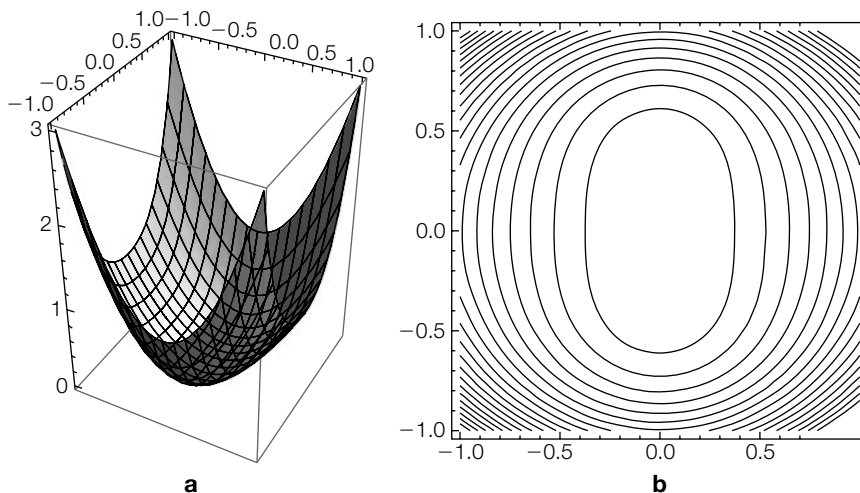


FIGURE 3.64

(a) Three-dimensional and (b) contour plots of $f(x, y)$

and parametric equations are

$$\begin{cases} x = x_0 + f_x(x_0, y_0) t \\ y = y_0 + f_y(x_0, y_0) t \\ z = z_0 - t. \end{cases} \quad (3.34)$$

The plane tangent to the graph of $z = f(x, y)$ at the point $(x_0, y_0, f(x_0, y_0))$ is the “best” linear approximation of $z = f(x, y)$ near $(x, y) = (x_0, y_0)$ in the same way as the line tangent to the graph of $y = f(x)$ at the point $(x_0, f(x_0))$ is the “best” linear approximation of $y = f(x)$ near $x = x_0$.

Example 3.5.7 Find an equation of the plane tangent and normal line to the graph of $f(x, y) = 4 - \frac{1}{4}(2x^2 + y^2)$ at the point $(1, 2, 5/2)$.

Solution We define $f(x, y)$ and compute $f_x(1, 2)$ and $f_y(1, 2)$.

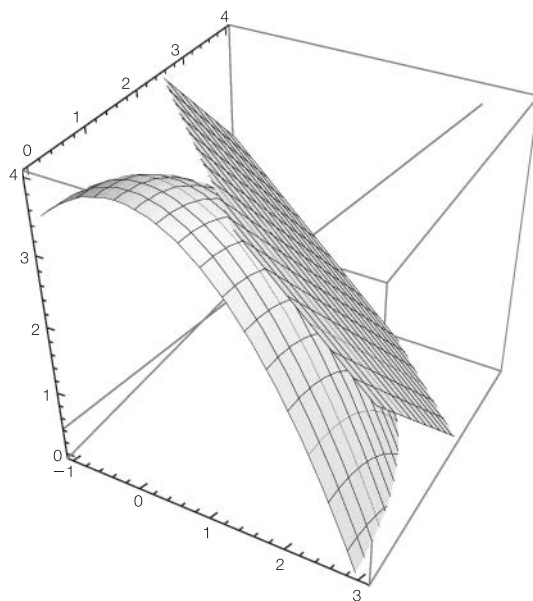
```
f[x_, y_] = 4 - 1/4(2x^2 + y^2);
f[1, 2]
dx = D[f[x, y], x]/.{x -> 1, y -> 2}
dy = D[f[x, y], y]/.{x -> 1, y -> 2}
5/2
-1
-1
```

Using (3.32), an equation of the tangent plane is $z = -1(x - 1) - 1(y - 2) + f(1, 2)$. Using (3.34), parametric equations of the normal line are $x = 1 - t$, $y = 2 - t$, $z = f(1, 2) - t$. We confirm the result graphically by graphing $f(x, y)$ together with the tangent plane in p1 using Plot3D. We use ParametricPlot3D to graph the normal line in p2 and then display p1 and p2 together with Show in Figure 3.65.

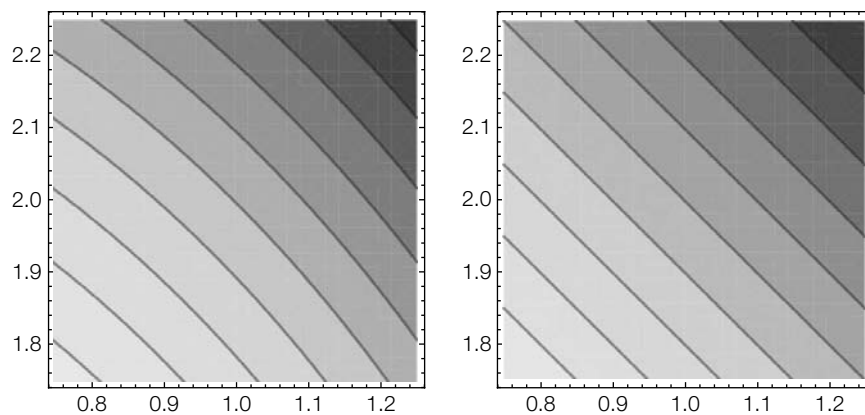
```
p1 = Plot3D[f[x, y], {x, -1, 3}, {y, 0, 4}];
p2 = Plot3D[dx (x - 1) + dy (y - 2) + f[1, 2], {x, -1, 3}, {y, 0, 4}];
p3 = ParametricPlot3D[{1 + dx t, 2 + dy t, f[1, 2] - t}, {t, -4, 4}];
Show[p1, p2, p3, PlotRange -> {{-1, 3}, {0, 4}, {0, 4}},
BoxRatios -> Automatic]
```

Because $z = -1(x - 1) - 1(y - 2) + f(1, 2)$ is the “best” linear approximation of $f(x, y)$ near $(1, 2)$, the graphs are very similar near $(1, 2)$ as shown in the three-dimensional plot. We also expect the level curves of each near $(1, 2)$ to be similar, which is confirmed with ContourPlot in Figure 3.66.

```
p4 = ContourPlot[f[x, y], {x, 0.75, 1.25}, {y, 1.75, 2.25}];
p5 = ContourPlot[dx(x - 1) + dy(y - 2) + f[1, 2], {x, 0.75, 1.25},
{y, 1.75, 2.25}];
Show[GraphicsRow[{p4, p5}]]
```

**FIGURE 3.65**

Graph of $f(x, y)$ with a tangent plane and normal line

**FIGURE 3.66**

Zooming in near $(1, 2)$

Lagrange Multipliers

Certain types of optimization problems can be solved using the method of *Lagrange multipliers* that is based on the following theorem.

Theorem 18 (Lagrange's Theorem). Let $z = f(x, y)$ and $z = g(x, y)$ be real-valued functions with continuous partial derivatives and let $z = f(x, y)$ have an extreme value at a point (x_0, y_0) on the smooth constraint curve $g(x, y) = 0$. If $\nabla g(x_0, y_0) \neq 0$, then there is a real number λ satisfying

$$\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0). \quad (3.35)$$

Graphically, the points (x_0, y_0) at which the extreme values occur correspond to the points where the level curves of $z = f(x, y)$ are tangent to the graph of $g(x, y) = 0$.

Example 3.5.8 Find the maximum and minimum values of $f(x, y) = xy$ subject to the constraint $\frac{1}{4}x^2 + \frac{1}{9}y^2 = 1$.

Solution For this problem, $f(x, y) = xy$ and $g(x, y) = \frac{1}{4}x^2 + \frac{1}{9}y^2 - 1$. Observe that parametric equations for $\frac{1}{4}x^2 + \frac{1}{9}y^2 = 1$ are $x = 2 \cos t$, $y = 3 \sin t$, $0 \leq t \leq 2\pi$. In Figure 3.67(a), we use **ParametricPlot3D** to parametrically graph $g(x, y) = 0$ and $f(x, y)$ for x - and y -values on the curve $g(x, y) = 0$ by graphing

$$\begin{cases} x = 2 \cos t \\ y = 3 \sin t \\ z = 0 \end{cases} \quad \text{and} \quad \begin{cases} x = 2 \cos t \\ y = 3 \sin t \\ z = x \cdot y = 6 \cos t \sin t \end{cases}$$

for $0 \leq t \leq 2\pi$. Our goal is to find the minimum and maximum values in Figure 3.67(a) and the points at which they occur.

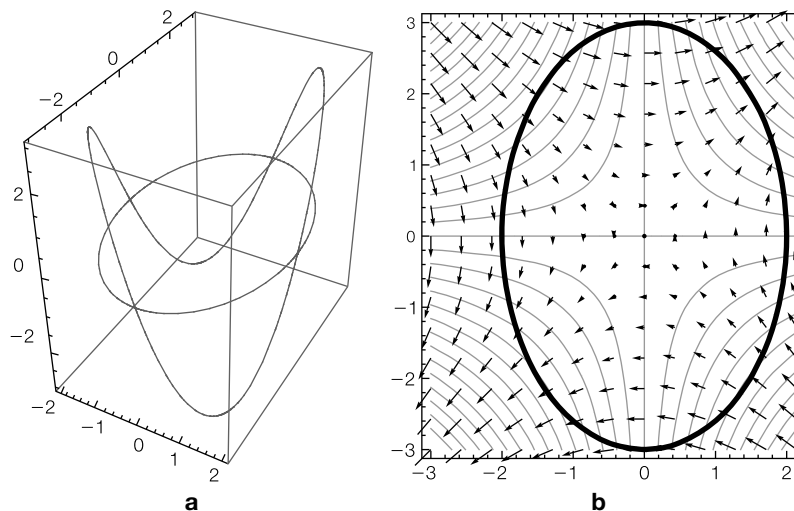


FIGURE 3.67

(a) $f(x, y)$ on $g(x, y) = 0$. (b) Level curves of $f(x, y)$ together with $g(x, y) = 0$

```

f[x_, y_] = xy;
g[x_, y_] = x^2/4 + y^2/9 - 1;
s1 = ParametricPlot3D[{2Cos[t], 3Sin[t], 0}, {t, 0, 2Pi}];
s2 = ParametricPlot3D[{2Cos[t], 3Sin[t], 6Cos[t]Sin[t]}, {t, 0, 2Pi}];
plot1 = Show[s1, s2, BoxRatios -> Automatic, PlotRange -> All];

```

To implement the method of Lagrange multipliers, we compute $f_x(x, y)$, $f_y(x, y)$, $g_x(x, y)$, and $g_y(x, y)$ with `D`.

```

fx = D[f[x, y], x]
fy = D[f[x, y], y]
gx = D[g[x, y], x]
gy = D[g[x, y], y]

```

$$\begin{matrix} y \\ x \\ \frac{x}{2} \\ \frac{2y}{9} \end{matrix}$$

`Solve` is used to solve the system of equations (3.35):

$$\begin{aligned} f_x(x, y) &= \lambda g_x(x, y) \\ f_y(x, y) &= \lambda g_y(x, y) \\ g(x, y) &= 0 \end{aligned}$$

for x , y , and λ .

```

vals = Solve[{fx==λgx, fy==λgy, g[x, y]==0}, {x, y, λ}]

```

$$\left\{ \left\{ \lambda \rightarrow -3, x \rightarrow -\sqrt{2}, y \rightarrow \frac{3}{\sqrt{2}} \right\}, \left\{ \lambda \rightarrow -3, x \rightarrow \sqrt{2}, y \rightarrow -\frac{3}{\sqrt{2}} \right\}, \right. \\ \left. \left\{ \lambda \rightarrow 3, x \rightarrow -\sqrt{2}, y \rightarrow -\frac{3}{\sqrt{2}} \right\}, \left\{ \lambda \rightarrow 3, x \rightarrow \sqrt{2}, y \rightarrow \frac{3}{\sqrt{2}} \right\} \right\}$$

The corresponding values of $f(x, y)$ are found using `ReplaceAll (/.)`.

```

n1 = {x, y, f[x, y]}/.vals//TableForm

```

$-\sqrt{2}$	$\frac{3}{\sqrt{2}}$	-3
$\sqrt{2}$	$-\frac{3}{\sqrt{2}}$	-3
$-\sqrt{2}$	$-\frac{3}{\sqrt{2}}$	3
$\sqrt{2}$	$\frac{3}{\sqrt{2}}$	3

```

N[n1]

```

-1.41421	2.12132	-3.
1.41421	-2.12132	-3.
-1.41421	-2.12132	3.
1.41421	2.12132	3.

We conclude that the maximum value $f(x, y)$ subject to the constraint $g(x, y) = 0$ is 3 and occurs at $\left(\sqrt{2}, \frac{3}{2}\sqrt{2}\right)$ and $\left(-\sqrt{2}, -\frac{3}{2}\sqrt{2}\right)$. The minimum value is -3 and

occurs at $(-\sqrt{2}, \frac{3}{2}\sqrt{2})$ and $(\sqrt{2}, -\frac{3}{2}\sqrt{2})$. We graph several level curves of $f(x, y)$ and the graph of $g(x, y) = 0$ with `ContourPlot` and show the graphs together with `Show`. The minimum and maximum values of $f(x, y)$ subject to the constraint $g(x, y) = 0$ occur at the points where the level curves of $f(x, y)$ are tangent to the graph of $g(x, y) = 0$ as illustrated in Figure 3.67(b).

```
<< "VectorFieldPlots"; gradfplot = PlotGradientField[f[x, y], {x, -3, 3},
{y, -3, 3}];
cp1 = ContourPlot[f[x, y], {x, -3, 3}, {y, -3, 3}, Contours->30,
ContourShading->False, PlotPoints->40];
cp2 = ContourPlot[g[x, y]==0, {x, -3, 3}, {y, -3, 3},
ContourStyle->Thickness[0.01],
ContourShading->False];
plot2 = Show[cp1, cp2, gradfplot];
Show[GraphicsRow[{plot1, plot2}]]
```

Observe that the maximum and minimum values occur where the gradient vectors of $z = f(x, y)$ are parallel to the gradient vectors of $z = g(x, y)$ on the equation $g(x, y) = 0$.

3.5.3 Iterated Integrals

The `Integrate` command, used to compute single integrals, is used to compute iterated integrals. The command

```
Integrate[f[x, y], {y, c, d}, {x, a, b}]
```

attempts to compute the iterated integral

$$\int_c^d \int_a^b f(x, y) dx dy. \quad (3.36)$$

If Mathematica cannot compute the exact value of the integral, it is returned unevaluated, in which case numerical results may be more useful. The iterated integral (3.36) is numerically evaluated with the command `N` or

```
NIntegrate[f[x, y], {y, c, d}, {x, a, b}]
```

Example 3.5.9 Evaluate each integral: (a) $\int_2^4 \int_1^2 (2xy^2 + 3x^2y) dx dy$; (b) $\int_0^2 \int_{y^2}^{2y} (3x^2 + y^3) dx dy$; (c) $\int_0^\infty \int_0^\infty xye^{-x^2-y^2} dy dx$; and (d) $\int_0^\pi \int_0^\pi e^{\sin xy} dx dy$.

Solution (a) First, we compute $\iint (2xy^2 + 3x^2y) dx dy$ with `Integrate`. Second, we compute $\int_2^4 \int_1^2 (2xy^2 + 3x^2y) dx dy$ with `Integrate`.

```
Integrate[2xy^2 + 3x^2y, y, x]
 $\frac{1}{6}x^2y^2(3x + 2y)$ 
```

Integrate[2xy^2 + 3x^2y, {y, 2, 4}, {x, 1, 2}]

98

(b) We illustrate the same commands as in (a), except we are integrating over a nonrectangular region.

Integrate[3x^2 + y^3, {x, y^2, 2y}]

$8y^3 + 2y^4 - y^5 - y^6$

Integrate[3x^2 + y^3, y, {x, y^2, 2y}]

$2y^4 + \frac{2y^5}{5} - \frac{y^6}{6} - \frac{y^7}{7}$

Integrate[3x^2 + y^3, {y, 0, 2}, {x, y^2, 2y}]

$\frac{1664}{105}$

(c) Improper integrals can be handled in the same way as proper integrals.

Integrate[xyExp[-x^2 - y^2], x, y]

$\frac{1}{4}e^{-x^2 - y^2}$

Integrate[xyExp[-x^2 - y^2], {x, 0, Infinity}, {y, 0, Infinity}]

$\frac{1}{4}$

(d) In this case, Mathematica cannot evaluate the integral exactly so we use **NIntegrate** to obtain an approximation.

Integrate[Exp[Sin[xy]], y, x]

$\int \int e^{\sin[xy]} dx dy$

NIntegrate[Exp[Sin[xy]], {y, 0, Pi}, {x, 0, Pi}]

15.5092

Area, Volume, and Surface Area

Typical applications of iterated integrals include determining the area of a planar region, the volume of a region in three-dimensional space, or the surface area of a region in three-dimensional space. The area of the planar region R is given by

$$A = \iint_R dA. \quad (3.37)$$

If $z = f(x, y)$ has continuous partial derivatives on a closed region R , then the surface area of the portion of the surface that projects onto R is given by

$$SA = \iint_R \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + 1} dA. \quad (3.38)$$

If $f(x, y) \geq g(x, y)$ on R , the volume of the region between the graphs of $f(x, y)$ and $g(x, y)$ is

$$V = \iint_R (f(x, y) - g(x, y)) dA. \quad (3.39)$$

Example 3.5.10 Find the area of the region R bounded by the graphs of $y = 2x^2$ and $y = 1 + x^2$.

Solution We begin by graphing $y = 2x^2$ and $y = 1 + x^2$ with **Plot** in Figure 3.68. The x -coordinates of the intersection points are found with **Solve**.

Plot[Tooltip[{ $2x^2$, $1 + x^2$ }], { x , $-3/2$, $3/2$ }]

Solve[$2x^2 == 1 + x^2$]

{ $\{x \rightarrow -1\}$, $\{x \rightarrow 1\}$ }

Using (3.37) and taking advantage of symmetry, the area of R is given by

$$A = \iint_R dA = 2 \int_0^1 \int_{2x^2}^{1+x^2} dy \, dx,$$

which we compute with **Integrate**.

2 Integrate[1, { x , 0, 1}, { y , $2x^2$, $1 + x^2$ }]

$\frac{4}{3}$

We conclude that the area of R is $4/3$.

If the problem exhibits “circular symmetry,” changing to polar coordinates is often useful. If $R = \{(r, \theta) | a \leq r \leq b, \alpha \leq \theta \leq \beta\}$, then

$$\iint_R f(x, y) \, dA = \int_{\alpha}^{\beta} \int_a^b f(r \cos \theta, r \sin \theta) r \, dr \, d\theta.$$

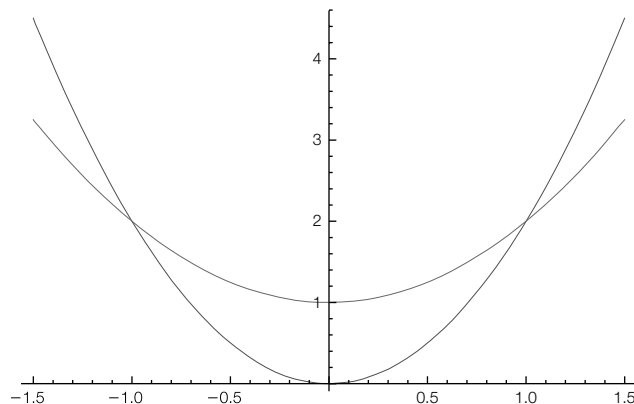


FIGURE 3.68

$y = 2x^2$ and $y = 1 + x^2$ for $-3/2 \leq x \leq 3/2$

Example 3.5.11 Find the surface area of the portion of

$$f(x, y) = \sqrt{4 - x^2 - y^2}$$

that lies above the region $R = \{(x, y) \mid x^2 + y^2 \leq 1\}$.

Solution First, observe that the domain of $f(x, y)$ is

$$\left\{ (x, y) \mid -\sqrt{4 - y^2} \leq x \leq \sqrt{4 - y^2}, -2 \leq y \leq 2 \right\} = \{(r, \theta) \mid 0 \leq r \leq 2, 0 \leq \theta \leq 2\pi\}.$$

Similarly,

$$R = \left\{ (x, y) \mid -\sqrt{1 - y^2} \leq x \leq \sqrt{1 - y^2}, -1 \leq y \leq 1 \right\} = \{(r, \theta) \mid 0 \leq r \leq 1, 0 \leq \theta \leq 2\pi\}.$$

With this observation, we use `ParametricPlot3D` to graph $f(x, y)$ in `p1` and the portion of the graph of $f(x, y)$ above R in `p2` and show the two graphs together with `Show`. We wish to find the area of the black region in Figure 3.69.

```
f[x_, y_] = Sqrt[4 - x^2 - y^2];
p1 = ParametricPlot3D[{r Cos[t], r Sin[t], f[r Cos[t], r Sin[t]]}, {r, 0, 2},
    {t, 0, 2Pi}, PlotPoints -> 45, ColorFunction -> "LightTerrain"];
p2 = ParametricPlot3D[{r Cos[t], r Sin[t], f[r Cos[t], r Sin[t]]}, {r, 0, 1},
```

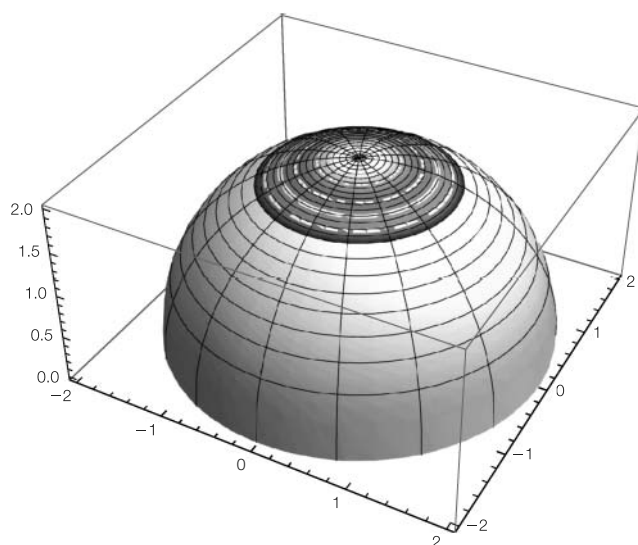


FIGURE 3.69

The portion of the graph of $f(x, y)$ above R

{t, 0, 2Pi}, PlotPoints → 45, ColorFunction → “DarkTerrain”];
Show[p1, p2, BoxRatios → Automatic]

We compute $f_x(x, y)$, $f_y(x, y)$ and $\sqrt{[f_x(x, y)]^2 + [f_y(x, y)]^2 + 1}$ with D and Simplify.

fx = D[f[x, y], x]

fy = D[f[x, y], y]

$$-\frac{x}{\sqrt{4-x^2-y^2}}$$

$$-\frac{y}{\sqrt{4-x^2-y^2}}$$

s1 = Simplify[Sqrt[1 + fx^2 + fy^2]]

$$2\sqrt{-\frac{1}{-4+x^2+y^2}}$$

Then, using (3.38), the surface area is given by

$$\begin{aligned} SA &= \iint_R \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + 1} dA \\ &= \iint_R \frac{2}{\sqrt{4-x^2-y^2}} dA \\ &= \int_{-1}^1 \int_{-\sqrt{1-y^2}}^{\sqrt{1-y^2}} \frac{2}{\sqrt{4-x^2-y^2}} dx dy. \end{aligned} \quad (3.40)$$

However, notice that in polar coordinates,

$$R = \{(r, \theta) | 0 \leq r \leq 1, 0 \leq \theta \leq 2\pi\},$$

so in polar coordinates the surface area is given by

$$SA = \int_0^{2\pi} \int_0^1 \frac{2}{\sqrt{4-r^2}} r dr d\theta,$$

s2 = Simplify[s1/.{x → r Cos[t], y → r Sin[t]}]

$$2\sqrt{\frac{1}{4-r^2}}$$

which is much easier to evaluate than (3.40). We evaluate the iterated integral with Integrate

s3 = Integrate[r s2, {t, 0, 2Pi}, {r, 0, 1}]

$$-4 \left(-2 + \sqrt{3} \right) \pi$$

N[s3]

$$3.36715$$

and conclude that the surface area is $(8 - 4\sqrt{3})\pi \approx 3.367$.

Example 3.5.12 Find the volume of the region between the graphs of $z = 4 - x^2 - y^2$ and $z = 2 - x$.

Solution We begin by graphing $z = 4 - x^2 - y^2$ and $z = 2 - x$ together with Plot3D in Figure 3.70(a).

```
p1 = Plot3D[{4 - x^2 - y^2, 2 - x}, {x, -2, 2}, {y, -2, 2},
  PlotRange -> {{-2, 2}, {-2, 2}, {-2, 4}}, BoxRatios -> Automatic];
```

The region of integration, R , is determined by graphing $4 - x^2 - y^2 = 2 - x$ in Figure 3.70(b).

```
p2 = ContourPlot[4 - x^2 - y^2 - (2 - x) == 0, {x, -2, 2}, {y, -2, 2},
  PlotPoints -> 50, Frame -> False, Axes -> Automatic,
  AxesOrigin -> {0, 0}];
Show[GraphicsRow[{p1, p2}]]
```

Another way to see the situation illustrated in Figure 3.70 is to use RegionPlot3D, which works in the same way as RegionPlot but in three dimensions. Completing the square shows us that

$$\begin{aligned} R &= \left\{ (x, y) \mid \left(x - \frac{1}{2} \right)^2 + y^2 \leq \frac{9}{4} \right\} \\ &= \left\{ (x, y) \mid \frac{1}{2} - \frac{1}{2}\sqrt{9 - 4y^2} \leq x \leq \frac{1}{2} + \frac{1}{2}\sqrt{9 - 4y^2}, -\frac{3}{2} \leq y \leq \frac{3}{2} \right\}. \end{aligned}$$

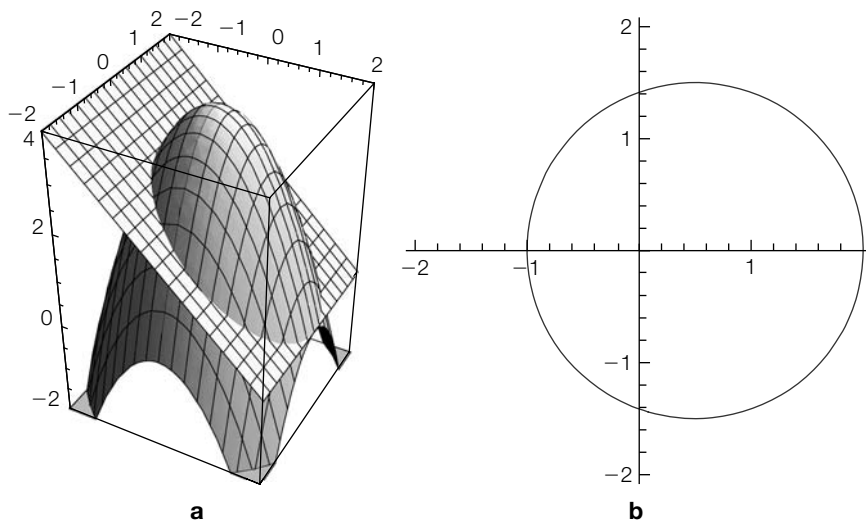
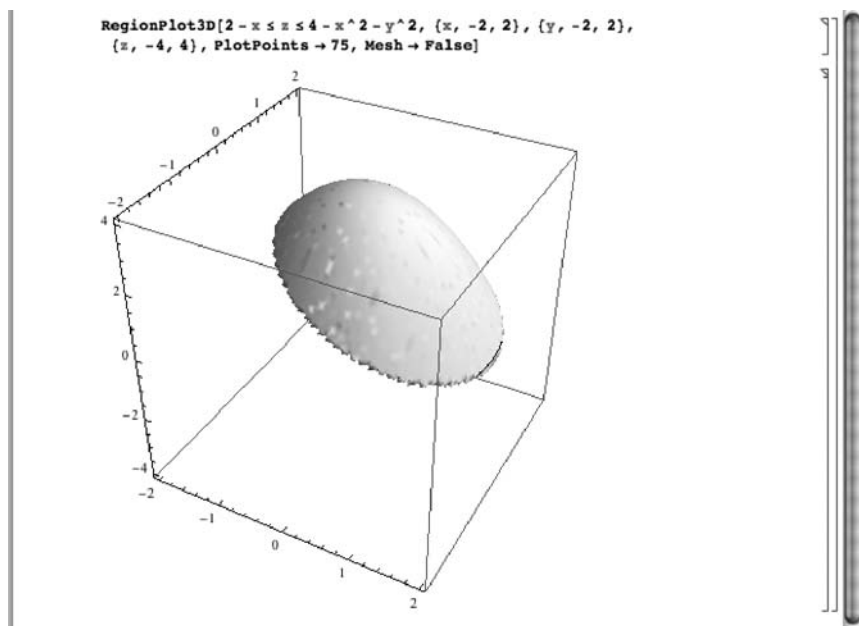


FIGURE 3.70

(a) $z = 4 - x^2 - y^2$ and $z = 2 - x$ for $-2 \leq x \leq 2$ and $-2 \leq y \leq 2$. (b) Graph of $4 - x^2 - y^2 = 2 - x$



Thus, using (3.39), the volume of the solid is given by

$$\begin{aligned}
 V &= \iint_R [(4 - x^2 - y^2) - (2 - x)] \, dA \\
 &= \int_{-\frac{3}{2}}^{\frac{3}{2}} \int_{\frac{1}{2} - \frac{1}{2}\sqrt{9-4y^2}}^{\frac{1}{2} + \frac{1}{2}\sqrt{9-4y^2}} [(4 - x^2 - y^2) - (2 - x)] \, dx \, dy,
 \end{aligned}$$

which we evaluate with Integrate.

```
i1 = Integrate[(4 - x^2 - y^2) - (2 - x), {y, -3/2, 3/2},
             {x, 1/2 - 1/2Sqrt[9 - 4y^2], 1/2 + 1/2Sqrt[9 - 4y^2]}]
```

```
 $\frac{81\pi}{32}$ 
```

```
N[i1]
```

```
7.95216
```

We conclude that the volume is $\frac{81}{32}\pi \approx 7.952$.

Triple Iterated Integrals

Triple iterated integrals are calculated in the same manner as double iterated integrals.

Example 3.5.13 Evaluate

$$\int_0^{\pi/4} \int_0^y \int_0^{y+z} (x + 2z) \sin y \, dx \, dz \, dy.$$

Solution Entering

```
i1 = Integrate[(x + 2z)Sin[y], {y, 0, Pi/4}, {z, 0, y}, {x, 0, y + z}]
- (17(384 - 96π - 12π² + π³))
  384√2
```

calculates the triple integral exactly with `Integrate`.

An approximation of the exact value is found with `N`.

```
N[i1]
0.157206
```

We illustrate how triple integrals can be used to find the volume of a solid when using spherical coordinates.

Example 3.5.14 Find the volume of the torus with equation in spherical coordinates $\rho = \sin \phi$.

Solution We proceed by graphing the torus with `SphericalPlot3D` in Figure 3.72 (see Figure 3.71 for the help feature associated with this command).

```
SphericalPlot3D[Sin[Phi], {Phi, 0, Pi}, {theta, 0, 2Pi}, PlotPoints -> 40]
```

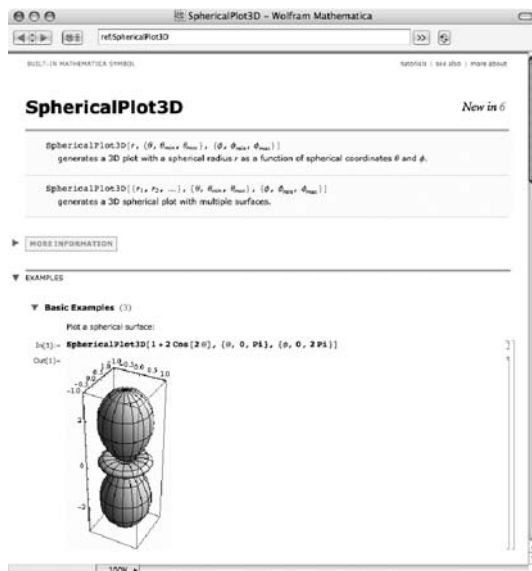
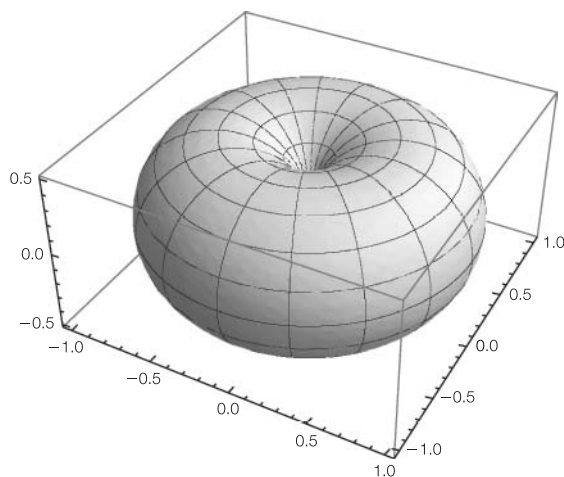


FIGURE 3.71

Mathematica's help for `SphericalPlot3D`

**FIGURE 3.72**

A graph of the torus

In general, the volume of the solid region D is given by

$$V = \iiint_D dV.$$

Thus, the volume of the torus is given by the triple iterated integral

$$V = \int_0^{2\pi} \int_0^\pi \int_0^{\sin \phi} \rho^2 \sin \phi \, d\rho \, d\phi \, d\theta,$$

```
i1 = Integrate[rho^2 Sin[phi], {theta, 0, 2Pi},
               {phi, 0, Pi}, {rho, 0, Sin[phi]}]
```

```
 $\frac{\pi^2}{4}$ 
```

```
N[i1]
```

```
2.4674
```

which we evaluate with `Integrate`. We conclude that the volume of the torus is $\frac{1}{4}\pi^2 \approx 2.467$.

3.6 EXERCISES

1. If \$ P is compounded n times per year at an annual interest rate of r , the value of the account, A , after t years is given by

$$A = \left(1 + \frac{r}{n}\right)^{nt}.$$

The formula for continuously compounded interest is obtained by finding the limit of this expression as $t \rightarrow \infty$. Find the limit.

2. Let $f(x) = \begin{cases} ax^4 + bx^3 + cx^2 + 8, & \text{if } x \leq z \\ ax^3 + bx^2 + cx + 4, & \text{if } x > z \end{cases}$.

(a) If $z = 2$, find a , b , and c so that $f(x)$, $f'(x)$, and $f''(x)$ are continuous for all real numbers.

(b) For what values of z , if any, are there no values a , b , and c so that $f(x)$, $f'(x)$, and $f''(x)$ are continuous for all real numbers?

3. Use Mathematica to generate a *representative* plot of each of the following functions. *Note:* If $x = p/q$ is rational, p/q is assumed to be a reduced fraction.

(a) $f(x) = \begin{cases} \ln p, & \text{if } x = p/q \text{ is rational;} \\ 0, & \text{if } x \text{ is irrational} \end{cases}$

(b) $f(x) = \begin{cases} \cos q, & \text{if } x = p/q \text{ is rational;} \\ 0, & \text{if } x \text{ is irrational} \end{cases}$

(c) $f(x) = \begin{cases} 1/p, & \text{if } x = p/q \text{ is rational;} \\ 0, & \text{if } x \text{ is irrational} \end{cases}$

(d) *Challenge:* Determine the value(s) of x , if any, for which each of these functions are continuous.

4. For $f(x) = \begin{cases} e^{-1/x^2}, & \text{if } x \neq 0 \\ 0, & \text{if } x = 0 \end{cases}$, provide a convincing argument that $f^{(n)}(0) = 0$ for all n . (Refer to Figure 3.3(b).)

5. Refer to Example 3.1.2. For what values of x , if any, is the function not continuous? Not differentiable?

6. (a) Find an equation of the line tangent to the graph of $f(x) = 9 - 4x^2$ at the point $(1, f(1))$. (b) Use Do to generate graphs of $y = f(x)$ and $y = f'(a)(x - a) + f(a)$ for 50 equally spaced values of a between -3 and 3 . (c) Use Table to create a similar plot for 9 equally spaced values of a between -3 and 3 and display the result as a graphics array.

7. Let $f(x) = mx + b$ and (x_0, y_0) be a point *not* on the graph of $f(x)$. Find the point on the graph of $f(x)$ that is closest to (x_0, y_0) .

8. If $f(x) = \cos(3x)/(x^2 + 1)$ on $[0, \pi]$, find the value(s) of c that satisfies that conclusion of the mean-value theorem for derivatives. Confirm your results graphically.

9. Sketch $f(x) = x^4 - x^2$. In your plot, label relative and absolute extreme values as well as points of inflection. *Tip:* A good plot indicates both the local and the global behavior of the function.

10. Use Maximize or Minimize to verify each of the results obtained in the examples in **Applied Max/Min Problems**, Section 3.2.6.

11. (a) Find dy/dx if $\cos(x + \sin y) = \sin y$. (b) Graph the equation for $-4\pi \leq x \leq 4\pi$ and $-4\pi \leq y \leq 4\pi$. (c) Find a point on the graph at which there are two tangents and then find equations of both tangents. (d) Illustrate that your final result is correct.
12. Find the ratio of the volume of the right circular cone of largest volume that can be circumscribed about a sphere of radius R to the volume of the right circular cone of largest volume that can be inscribed in a sphere of radius R .
13. Plot $f(x) = x(x-1)^{1/3}(x-2)^{2/3}$ without loading the **RealOnly** package. Calculate and then plot $f'(x)$ and $f''(x)$ as well. *Hint:* Use Abs.
14. Calculate (a) $\int \frac{1}{\sin^2 x + 2} dx$ and (b) $\int x^2 \tan^{-1} x dx$. In each case, check that your answer is correct by computing the antiderivative by hand.
15. Refer to Figure 3.30. Create a Manipulate object that can be used to illustrate how the lengths of the stayed wires change as D , L_1 , and L_2 change. Use $[0, 100]$ for each range. For the initial values set $D = 50$, $L_1 = 20$, and $L_2 = 60$.
16. Let $f(x) = ax^2 + c$. For $x = x_0$, let $L(x_0)$ denote the line perpendicular to the tangent at $(x_0, f(x_0))$ and let d denote the length of the line segment formed by the intersection $L(x_0)$ and $f(x)$. Find x_0 so that d is minimized. What is the measure of the angle formed by the intersection of the two lines for which d is minimized?
17. (a) Define functions `simpson`, which implements Simpson's rule, and `trapezoid`, which implements the trapezoidal rule by adjusting the function `leftsum` (middlesum or rightsum) discussed previously to perform the calculation for the desired method. (b) Let $f(x) = e^{-(x-3)^2 \cos(4(x-3))}$. (i) Graph $y = f(x)$ on the interval $[1, 5]$. Use (ii) Simpson's rule with $n = 4$, (iii) the trapezoidal rule with $n = 4$, and (iv) the midpoint rule with $n = 4$ to approximate $\int_1^5 f(x) dx$.
18. If $p(x) = \frac{3}{10}x^5 - 3x^4 + 11x^3 - 18x^2 + 12x + 1$ and $q(x) = -4x^3 + 28x^2 - 56x + 32$, find the solutions of $p(x) = q(x)$ using FindRoot. *Challenge:* Use Map together with FindRoot to perform the operation in a single command.
19. Let $f(x) = \exp\left(-(x-2)^2 \cos \pi x\right)$ and $g(x) = 4 \cos(x-2)$. (a) Find the area of the region bounded by the graphs of the two functions. (b) Find the volume of the solid obtained by revolving the region bounded by the graphs of the two functions about the x -axis. (c) Find the volume of the solid obtained by revolving the region bounded by the graphs of the two functions about the y -axis. (d) Generate plots illustrating the area and the two solids. *Hint:* Use FindRoot.
20. Let R denote the region in the first quadrant bounded by the graphs of $y = x^n$ and $x = y^n$. (a) Find the area of R . (b) Find the volume of the

- solid obtained by revolving R about the x -axis. (c) Find the volume of the solid obtained by revolving R about the y -axis.
21. Calculate (a) $\int_1^n k 2^{-k} dk$, (b) $\lim_{n \rightarrow \infty} \int_1^n k 2^{-k} dk$, and (c) $\int_1^\infty k 2^{-k} dk$.
22. Show that $\sum_{k=1}^\infty \frac{50^k}{k!}$ converges and find its sum.
23. Find $\sum_{k=1}^\infty x^{3k}$. What is the interval of convergence for this series?
24. Evaluate $\sum_{n=1}^\infty \frac{3^{n/2}}{5^n}$. Confirm your result by showing that the series converges and finding its sum by hand.
25. (a) Plot (k, a_k) for $k = 1, 2, \dots, 2000$ if $a_k = \frac{\sin k}{k}$. (b) Find $\sum_{k=1}^\infty \frac{\sin k}{k}$. *Challenge:* Prove that the infinite series converges. (c) Plot (k, a_k) for $k = 1, 2, \dots, 2000$ if $a_k = \frac{\sin k}{k^2}$. (d) Prove that the infinite series converges. *Challenge:* Find $\sum_{k=1}^\infty \frac{\sin k}{k^2}$. (See Figure 3.73(a).)
26. Let $f(x) = \exp(-(x-1)^2(x+1)^2)$. (a) Graph $f(x)$ together with its 8th degree Taylor polynomial expanded about $x = 1$ on the interval $[-1.75, 1.75]$. (b) What is the interval of convergence for the Taylor series about $x = 1$ for this function? (c) Can you use a Taylor polynomial expanded about $x = 1$ for this function to approximate $f(0)$? Explain. (See Figure 3.73(b).)
27. Find the length of the graph of $f(x) = \sin(x + x \sin x)$ from $x = 0$ to $x = 2\pi$.
28. Determine $\lim_{(x,y) \rightarrow (0,0)} xy/(x^2 + y^2)$. If the limit does not exist, confirm your results graphically.
29. *Minimal surfaces* have “zero mean curvature.” Minimal surfaces that are parametrically defined by $x = u$, $y = v$, $z = f(u, v)$ satisfy

See Chapter 5 for more discussion regarding curvature.

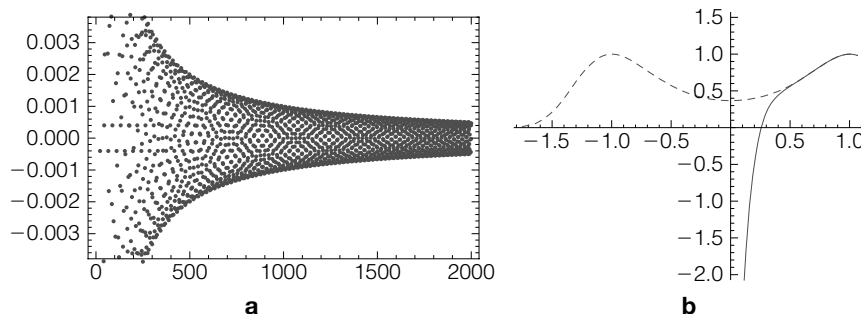


FIGURE 3.73

(a) The first 2000 terms of an interesting sequence. (b) What is the radius of the interval of convergence?

Lagrange's equation, $(1 + f_v^2)f_{uu} - 2f_u f_{uv} + (1 + f_u^2)f_{vv} = 0$. Plot

Enneper's minimal surface, $x = u - \frac{1}{3}u^3 + uv^2$, $y = v - \frac{1}{3}v^3 + u^2v$, $z = u^2 - v^2$, and show that the equations satisfy Lagrange's equation.

30. Let $g(x, y) = \exp\left(-\frac{1}{8}(x^2 + y^2)\right)(\cos^2 x + \sin^2 y)$. (a) Graph $g(x, y)$ using your favorite color scheme from the **Color Schemes** palette for $-\pi \leq x \leq \pi$ and $-\pi \leq y \leq \pi$. (b) Compute and simplify f_x , f_y , f_{xy} , f_{xx} , and f_{yy} using D and Simplify. (c) Use Mathematica help to determine the functionality of Derivative and then use Derivative to recalculate the partials.
31. Find and classify the critical points of $f(x, y) = -120x^3 - 30x^4 + 18x^5 + 5x^6 + 30xy^2$. Confirm your results with three-dimensional and contour plots with a gradient plot.
32. Find equations of the tangent plane and normal line to $f(x, y) = 4(x^2 + y^2 + 1)^{-1}$ at $(1/2, 1, f(1/2, 1))$. Confirm your results graphically. (See Figure 3.74(a).)
33. Find the minimum and maximum values of $f(x, y) = x^2 + 4y^3$ subject to $x^2 + 4y^2 = 1$. Confirm your results graphically. (See Figure 3.74(b).)
34. Evaluate $\int_1^2 \int_{1-y}^{\sqrt{y}} xy^2 dy dx$.
35. Evaluate $\int_0^{\sqrt{\pi}} \int_0^{\sqrt{\pi}} \cos(x^2 - y^2) dy dx$. Determine the meaning of the functions FresnelC and FresnelS.

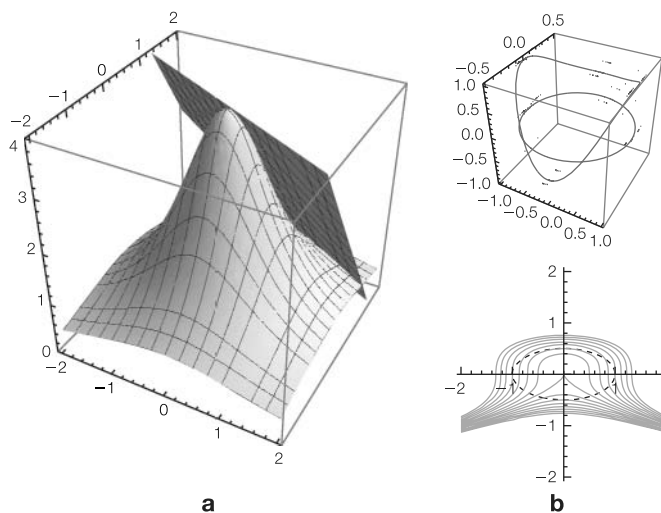


FIGURE 3.74

(a) Tangent plane. (b) Lagrange multipliers

Introduction to Lists and Tables

Chapter 4 introduces operations on lists and tables. The examples used to illustrate the various commands in this chapter are taken from calculus, business, dynamical systems, and engineering applications.

4.1 LISTS AND LIST OPERATIONS

4.1.1 Defining Lists

A **list** of n elements is a Mathematica object of the form

$$\text{list} = \{a_1, a_2, a_3, \dots, a_n\}$$

The i th element of the list is extracted from list with `list[[i]]` or `Part[list,i]`.

Elements of a list are separated by commas. Lists are always enclosed in braces $\{\dots\}$ and each element of a list may be (almost any) Mathematica object—even other lists. Because lists are Mathematica objects, they can be named. For easy reference, we will usually name lists.

Lists can be defined in a variety of ways: They may be completely typed in, imported from other programs and text files, or they may be created with either the `Table` or `Array` commands. Given a function $f(x)$ and a number n , the command

1. `Table[f[i],{i,n}]` creates the list $\{f[1], \dots, f[n]\}$;
2. `Table[f[i],{i,0,n}]` creates the list $\{f[0], \dots, f[n]\}$;
3. `Table[f[i],{i,n,m}]` creates the list

$$\{f[n], f[n+1], \dots, f[m-1], f[m]\};$$

4. `Table[f[i],{i,imin,imax,istep}]` creates the list

$$\{f[imin], f[imin+istep], f[imin+2*step], \dots, f[imax]\};$$

and

Table and Manipulate have nearly identical syntax. With Manipulate, you can create an interactive dynamic application; Table returns nonadjustable results.

5. `Array[f,n]` creates the list $\{f[1], \dots, f[n]\}$.

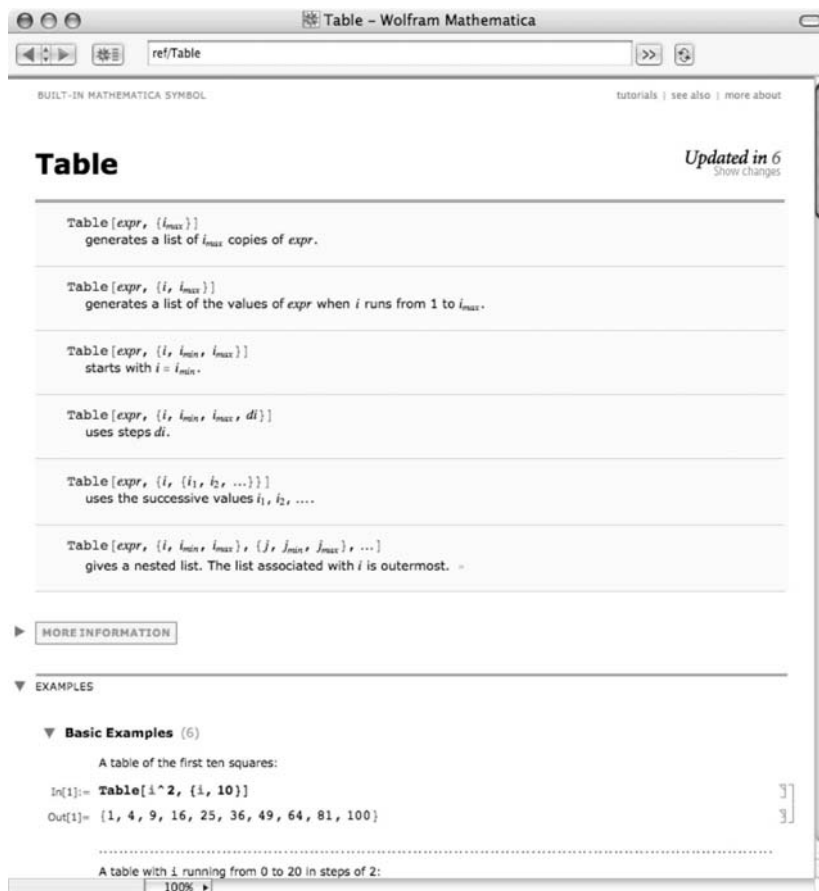
In particular,

`Table[f[x], {x, a, b, (b-a)/(n-1)}]`

returns a list of $f(x)$ values for n equally spaced values of x between a and b ;

`Table[{x, f[x]}, {x, a, b, (b-a)/(n-1)}]`

returns a list of points $(x, f(x))$ for n equally spaced values of x between a and b .



In addition to using `Table`, lists of numbers can be calculated using `Range`:

1. `Range[n]` generates the list $\{1, 2, \dots, n\}$;

2. `Range[n1,n2]` generates the list $\{n1, n1+1, \dots, n2-1, n2\}$; and
3. `Range[n1,n2,nstep]` generates the list

$$\{n1, n1 + nstep, n1 + 2*nstep, \dots, n2 - nstep, n2\}$$

Example 4.1.1 Use Mathematica to generate the list $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

Solution Generally, a given list can be constructed in several ways. In fact, each of the following five commands generates the list $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

`{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}`
 $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

`Table[i, {i, 10}]`
 $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

`Table[i, {i, 1, 10}]`
 $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

`Table[ $\frac{i}{2}$ , {i, 2, 20, 2}]`
 $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

`Range[10]`
 $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

Example 4.1.2 Use Mathematica to define `listone` to be the list of numbers $\{1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4\}$.

Solution In this case, we generate a list and name the result `listone`. As in Example 4.1.1, we illustrate that `listone` can be created in several ways.

`listone = {1,  $\frac{3}{2}$ , 2,  $\frac{5}{2}$ , 3,  $\frac{7}{2}$ , 4}`
 $\{1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4\}$

`listone = Table[i, {i, 1, 4,  $\frac{1}{2}$ }]`
 $\{1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4\}$

Last, we define $i(n) = \frac{1}{2}n + \frac{1}{2}$ and use `Array` to create the table `listone`.

`i[n_] =  $\frac{n}{2} + \frac{1}{2}$ ;`
`listone = Array[i, 7]`
 $\{1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4\}$

Example 4.1.3 Create a list of the first 25 prime numbers. What is the 15th prime number?

Solution The command `Prime[n]` yields the n th prime number. We use `Table` to generate a list of the ordered pairs $\{n, \text{Prime}[n]\}$ for $n = 1, 2, 3, \dots, 25$ and name the resulting list

list. We then use **Short** to obtain an abbreviated portion of **list**. Generally, **Short** returns the first and last few elements of a list. The number of omitted terms between the first few and last few is indicated with `<<n>>`. In this case, we see that 17 terms are omitted.

```
list = Table[{n, Prime[n]}, {n, 1, 25}];
Short[list]
{{1, 2}, {2, 3}, {3, 5}, {4, 7},
<<17>>, {22, 79}, {23, 83}, {24, 89}, {25, 97}}
```

The i th element of a list **list** is extracted from **list** with **list[[i]]** or **Part[list,i]**. From the resulting output, we see that the 15th prime number is 47.

```
list[[15]]
{15, 47}

Part[list, 15]
{15, 47}
```

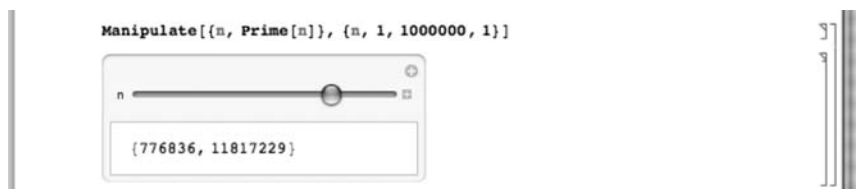
You can use the **Manipulate** function in nearly the exact same way as the **Table** function. With **Manipulate**, the result is an interactive dynamic object that can be saved as an application that can be run outside of Mathematica. With



we let i and $i/2$ vary continuously for $1 \leq i \leq 10$. By making the stepsize be 1, integer values of i are only allowed.



With the following **Manipulate** command, you can see n and the n th prime number for $1 \leq n \leq 1000000$.



In addition, we can use `Table` to generate lists consisting of the same or similar objects.

Example 4.1.4 (a) Generate a list consisting of five copies of the letter *a*. (b) Generate a list consisting of 10 random integers between -10 and 10 and then a list of 10 random real numbers between -10 and 10 .

Solution Entering

```
Table[a, {5}]
```

```
{a, a, a, a, a}
```

generates a list consisting of five copies of the letter *a*. For (b), we use the commands `RandomInteger` and `RandomReal` to generate the desired lists. Because we are using `RandomInteger` and `RandomReal`, your results will certainly differ from those obtained here.

```
RandomInteger[{-10, 10}, 10]
```

```
{3, -5, 5, -8, 0, -2, -2, 2, 7, 9}
```

```
RandomReal[{-10, 10}, 10]
```

```
{-3.42641, 4.76027, -3.49249, -9.11795, 3.72502, 7.39518,  
-6.84238, -7.85735, 4.94279, -9.4021}
```

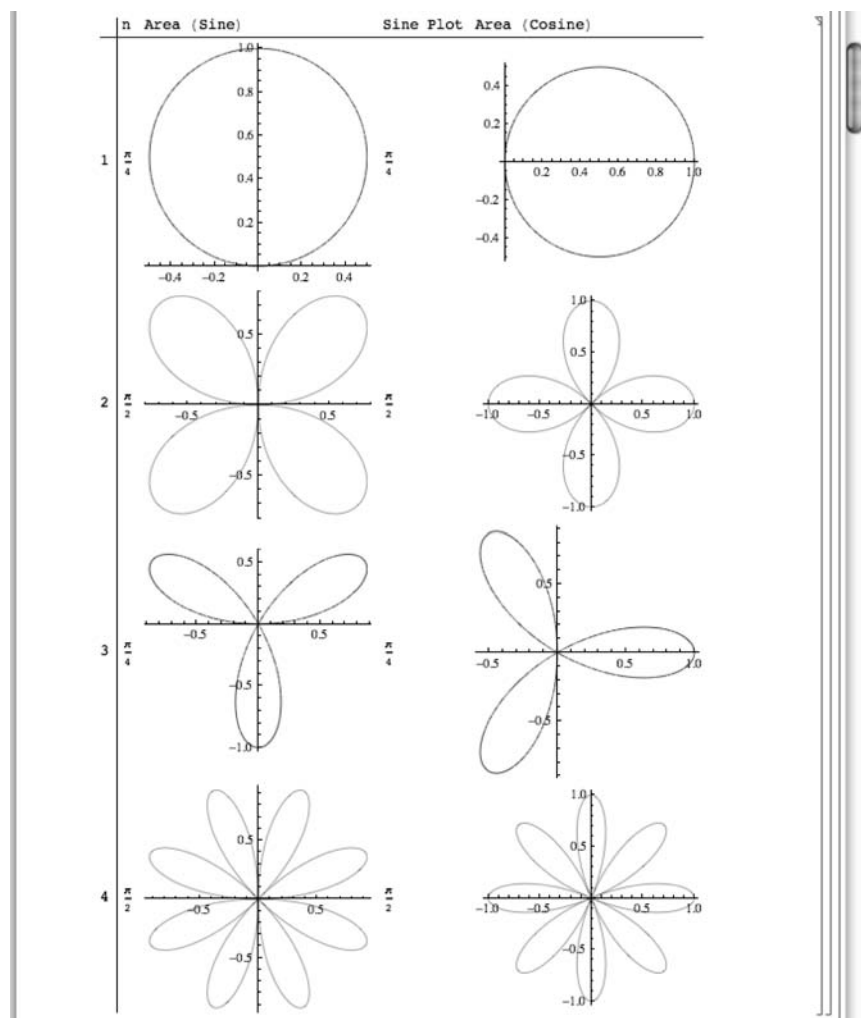
As illustrated previously, `Manipulate` works in much the same way as `Table` but allows you to interactively see how adjusting parameters affects a given situation.

Example 4.1.5 In polar coordinates, the graphs of $r = \sin n\theta$ and $r = \cos n\theta$ are n -leaved roses if n is odd and $2n$ -leaved roses if n is even. If n is even, the area of the graph enclosed by the $2n$ roses is $A = \frac{1}{2} \int_0^{2\pi} r^2 d\theta = \pi/2$. If n is odd, the area of the graph enclosed by the n roses is $A = \frac{1}{2} \int_0^{2\pi} r^2 d\theta = \pi/4$.

To see this with Mathematica, we can use `Table`. (See Figure 4.1.) (Note that `If[condition,f,g]` returns *f* if *condition* is `True` and *g* if it is not.)

```
Clear[n, x];
```

```
t1 = Table[{
```

**FIGURE 4.1**

You can use Table to see that the area of the roses depends only on whether n is odd or even

```

If[Mod[n/2, 1] === 0, Integrate[Sin[nx]^2, {x, 0, 2Pi}]/2,
  Integrate[Sin[nx]^2, {x, 0, Pi}]/2,
  PolarPlot[Sin[nx], {x, 0, 2Pi}],
If[Mod[n/2, 1] === 0, Integrate[Cos[nx]^2, {x, 0, 2Pi}]/2,
  Integrate[Cos[nx]^2, {x, 0, Pi}]/2,
  PolarPlot[Cos[nx], {x, 0, 2Pi}]], {n, 1, 5}];

```

```
TableForm[t1,
  TableHeadings → {Table[n, {n, 1, 5}],
    {"n", "Area(Sine)", "SinePlot", "Area(Cosine)", "CosinePlot"}}]
```

Alternatively, you can use Manipulate. (See Figure 4.2.)

```
Clear[n, x];
Manipulate[{n,
  If[Mod[n/2, 1] === 0, Integrate[Sin[nx]^2, {x, 0, 2Pi}]/2,
    Integrate[Sin[nx]^2, {x, 0, Pi}]/2,
    PolarPlot[Sin[nx], {x, 0, 2Pi}],
  If[Mod[n/2, 1] === 0, Integrate[Cos[nx]^2, {x, 0, 2Pi}]/2,
    Integrate[Cos[nx]^2, {x, 0, Pi}]/2,
    PolarPlot[Cos[nx], {x, 0, 2Pi}]], {n, 5, 1, 100, 1}]
```

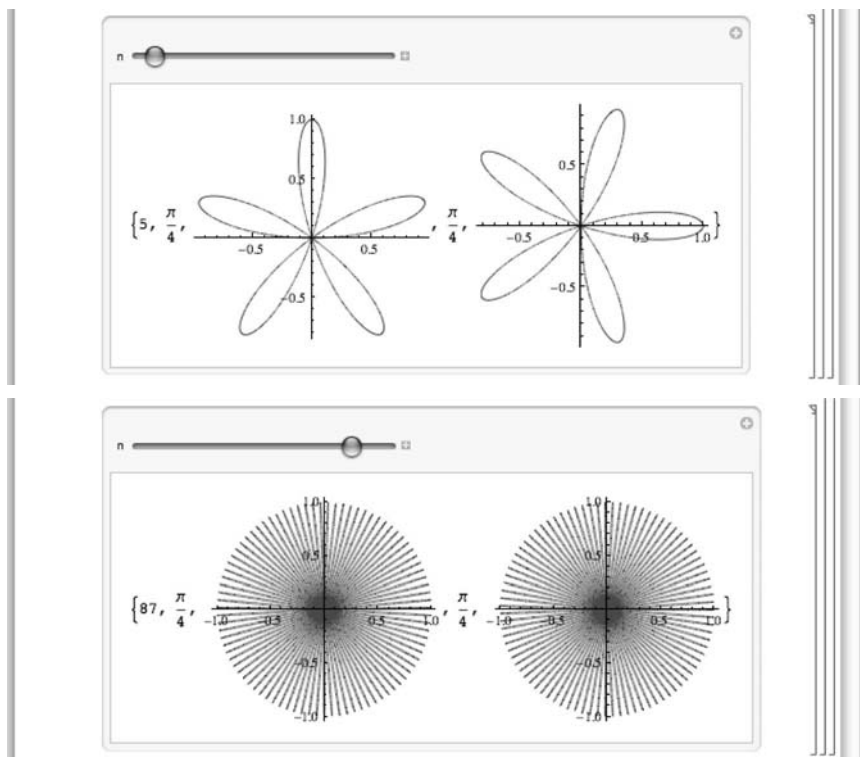


FIGURE 4.2

With Manipulate you can see that the area alternates from $\pi/2$ to $\pi/4$ as n alternates from even to odd

4.1.2 Plotting Lists of Points

Lists are plotted with `ListPlot`.

1. `ListPlot[{{x1,y1},{x2,y2},...,{xn,yn}}]` plots the list of points $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. The size of the points in the resulting plot is controlled with the option `PlotStyle->PointSize[w]`, where w is the fraction of the total width of the graphic. For two-dimensional graphics, the default value is 0.008.
2. `ListPlot[{y1,y2,...,yn}]` plots the list of points $\{(1, y_1), (2, y_2), \dots, (n, y_n)\}$.

To connect the consecutive points with line segments, use the option `Joined->True`.

Example 4.1.6 Entering

When a semicolon is included at the end of a command, the resulting output is suppressed.

```
t1 = Table[Sin[n], {n, 1, 1000}];  
ListPlot[t1]
```

creates a list consisting of $\sin n$ for $n = 1, 2, \dots, 1000$ and then graphs the list of points $(n, \sin n)$ for $n = 1, 2, \dots, 1000$. See Figure 4.3.

Example 4.1.7 (The Prime Difference Function and the Prime Number Theorem).

In `t1`, we use `Prime` and `Table` to compute a list of the first 25,000 prime numbers.

```
t1 = Table[Prime[n], {n, 1, 25000}];
```

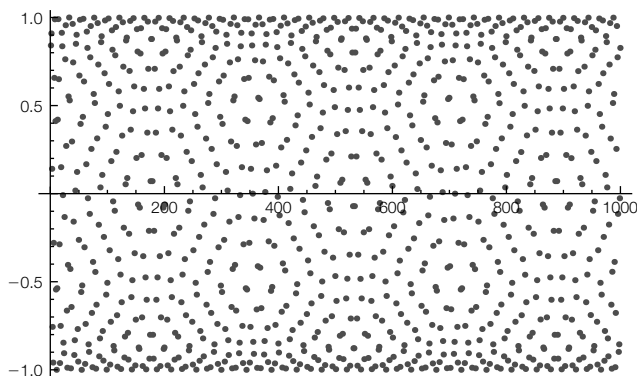


FIGURE 4.3

Plot of $(n, \sin n)$ for $n = 1, 2, \dots, 1000$

We use **Length** to verify that **t1** has 25,000 elements and **Short** to see an abbreviated portion of **t1**.

Length[t1]

25000

Short[t1]

{2, 3, 5, 7, 11, 13, 17, 19, <{24984}>, 287047,
287057, 287059, 287087, 287093, 287099, 287107, 287117}

You can also use **Take** to extract elements of lists.

First[list] returns the first element of list;
Last[list] returns the last element of list.

1. **Take[list,n]** returns the first *n* elements of list;
2. **Take[list,-n]** returns the last *n* elements of list; and
3. **Take[list,{n,m}]** returns the *n*th through *m*th elements of list.

Take[t1,5]

{2, 3, 5, 7, 11}

Take[t1, -5]

{287087, 287093, 287099, 287107, 287117}

Take[t1,{12501,12505}]

{134059, 134077, 134081, 134087, 134089}

Span is new in Mathematica 6 but works in almost the same way as **Take**.

Working in almost the same way as **Take**, **Span (;)** selects elements of lists: **list[[n;;m]** returns the *n* through *m*th elements of list.

Example 4.1.8 Here are the first few terms of sequence A073184,¹ the number of cube free divisors of *n*:

**ashortlist = {1, 2, 2, 3, 2, 4, 2, 3, 3, 4, 2, 6,
2, 4, 4, 3, 2, 6, 2, 6, 4, 4, 2, 6,
3, 4, 3, 6, 2, 8, 2, 3, 4, 4, 4};**

With **;;** (**Span**), we select the second through eighth elements of **ashortlist**.

ashortlist[[2;;8]]

{2, 2, 3, 2, 4, 2, 3}

The same results are obtained with **Take**.

Take[ashortlist, {2, 8}]

{2, 2, 3, 2, 4, 2, 3}

You can count the number of elements of a list with **Length**.

Length[ashortlist]

35

¹ Sloane, N. J. A., *The On-Line Encyclopedia of Integer Sequences*, www.research.att.com/njas/sequences, 2007.

With **Tally**, we count the number of occurrences of each digit in the list. Thus,

```
Tally[ashortlist]
{{1, 1}, {2, 11}, {3, 7}, {4, 10}, {6, 5}, {8, 1}}
```

shows us that there are eleven 2's, ten 4's, and so on.

However, you can use **Table** together with **Part** (**[...]**) to obtain the same results as those obtained with **Take** or **Span**.

```
Table[t1[[i]], {i, 1, 5}]
Table[t1[[i]], {i, 24996, 25000}]
Table[t1[[i]], {i, 12501, 12505}]
{2, 3, 5, 7, 11}
{287087, 287093, 287099, 287107, 287117}
{134059, 134077, 134081, 134087, 134089}
```

In **t2**, we compute the difference, d_n , between the successive prime numbers in **t1**. The result is plotted with **ListPlot** in Figure 4.4.

list[[i]] returns the i th element of list so **list[[i+1]] - list[[i]]** computes the difference between the $(i+1)$ st and i th elements of list.

```
t2 = Table[t1[[i + 1]] - t1[[i]], {i, 1, Length[t1] - 1};
Short[t2]
{1, 2, 2, 4, 2, 4, 2, 4, 6, 2, 6, 4, 2, 4, 6, 6, <24967>,
28, 14, 54, 46, 8, 6, 12, 4, 44, 10, 2, 28, 6, 6, 8, 10}
ListPlot[t2, PlotRange -> All]
```

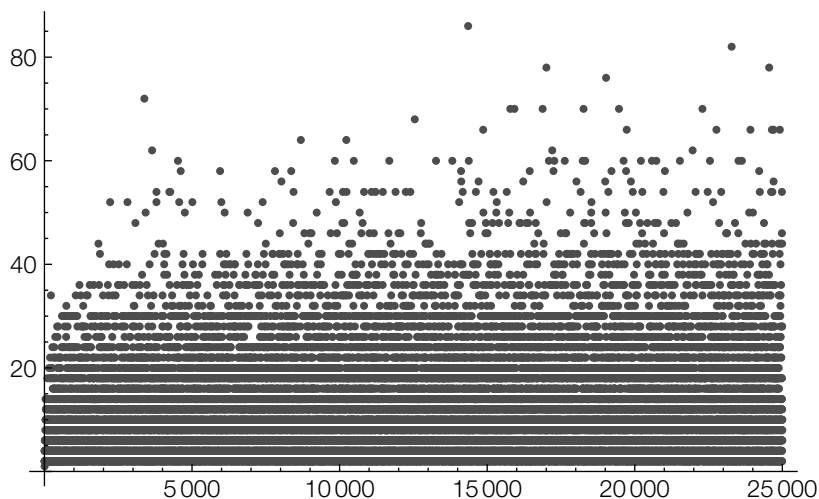


FIGURE 4.4

A plot of the difference, d_n , between successive prime numbers

Let $\pi(n)$ denote the number of primes less than n and $Li(x)$ denote the **logarithmic integral**:

$$\text{LogIntegral}[x] = Li(x) = \int_0^x \frac{1}{\ln t} dt.$$

Remember that p1 is not displayed because a semicolon is included at the end of the Plot command.

We use Plot to graph $Li(x)$ for $1 \leq x \leq 25,000$ in p1.

p1 = Plot[LogIntegral[x], {x, 1, 2500}];

The **prime number theorem** states that

$$\pi(n) \sim Li(n).$$

(See [20].) In the following, we use **Select** and **Length** to define $\pi(n)$. **Select[list,criteria]** returns the elements of **list** for which **criteria** is true. Note that $\#<n$ is called a **pure function**: Given an argument $\#$, $\#<n$ is true if $\#<n$ and false otherwise. The **&** symbol marks the end of a pure function. Thus, given n , **Select[t1,#<n&]** returns a list of the elements of **t1** less than n ; **Select[t1,#<n&]/Length** returns the number of elements in the list.

smallpi[n_] := Select[t1, # < n&] / Length

For example,

smallpi[100]
25

shows us that $\pi(100) = 25$. Note that because **t1** contains the first 25,000 primes, **smallpi[n]** is valid for $1 \leq n \leq N$, where $\pi(N) = 25,000$. In **t3**, we compute $\pi(n)$ for $n = 1, 2, \dots, 25,000$

t3 = Table[smallpi[n], {n, 1, 2500}];
Short[t3]
{0, 0, 1, 2, 2, 3, 3, 4, 4, 4, 5, 5, <<2475>>,
367, 367, 367, 367, 367, 367, 367, 367, 367, 367, 367, 367}

and plot the resulting list with **ListPlot**.

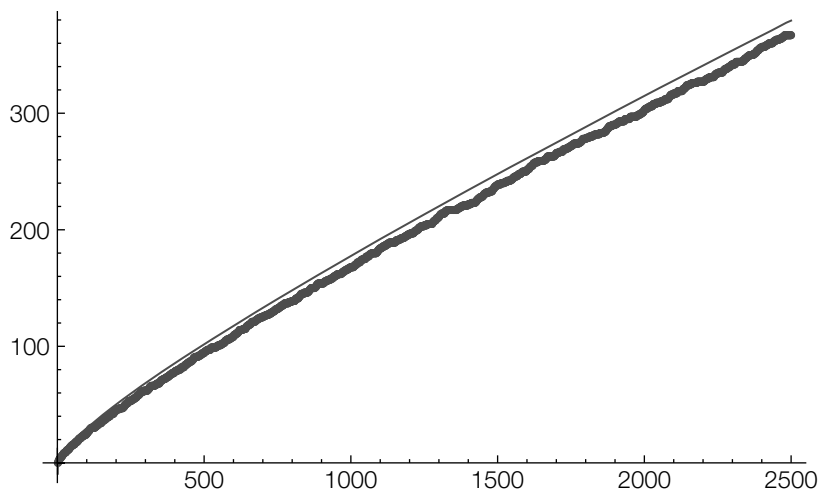
p2 = ListPlot[t3, PlotStyle -> GrayLevel[0.4]]

p1 and p2 are displayed together with **Show** in Figure 4.5.

Show[p1, p2]

You can iterate recursively with **Table**. Both

t1 = Table[a[i, j], {j, 2, 10, 2}, {i, 1, 5}]
{ {a[1, 2], a[2, 2], a[3, 2], a[4, 2], a[5, 2]},
{a[1, 4], a[2, 4], a[3, 4], a[4, 4], a[5, 4]}, {a[1, 6], a[2, 6], a[3, 6], a[4, 6], a[5, 6]},
{a[1, 8], a[2, 8], a[3, 8], a[4, 8], a[5, 8]}, {a[1, 10], a[2, 10], a[3, 10],
a[4, 10], a[5, 10]}}
Length[t1]
5

**FIGURE 4.5**

Graphs of $Li(x)$ (in black) and $\pi(n)$ (in gray)

and

```
t2 = Table[Table[a[i, j], {i, 1, 5}], {j, 2, 10, 2}]
{{a[1, 2], a[2, 2], a[3, 2], a[4, 2], a[5, 2]}, {a[1, 4], a[2, 4], a[3, 4], a[4, 4], a[5, 4]},
{a[1, 6], a[2, 6], a[3, 6], a[4, 6], a[5, 6]}, {a[1, 8], a[2, 8], a[3, 8], a[4, 8], a[5, 8]},
{a[1, 10], a[2, 10], a[3, 10], a[4, 10], a[5, 10]}}
```

compute tables of a_{ij} . The outermost iterator is evaluated first: In this case, i is followed by j as in $t1$ and the result is a list of lists. To eliminate the inner lists (that is, the braces), use `Flatten`. Generally, `Flatten[list, n]` flattens list (removes braces) to level n .

```
Flatten[t1]
{a[1, 2], a[2, 2], a[3, 2], a[4, 2], a[5, 2], a[1, 4], a[2, 4], a[3, 4], a[4, 4],
a[5, 4], a[1, 6], a[2, 6], a[3, 6], a[4, 6], a[5, 6], a[1, 8], a[2, 8], a[3, 8], a[4, 8], a[5, 8],
a[1, 10], a[2, 10], a[3, 10], a[4, 10], a[5, 10]}
```

The observation is especially important when graphing lists of points obtained by iterating `Table`. For example,

```
t1 = Table[{Sin[x + y], Cos[x - y]}, {x, 1, 5}, {y, 1, 5}]
{{{Sin[2], 1}, {Sin[3], Cos[1]}, {Sin[4], Cos[2]}, {Sin[5], Cos[3]}, {Sin[6], Cos[4]}},
{{Sin[3], Cos[1]}, {Sin[4], 1}, {Sin[5], Cos[1]}, {Sin[6], Cos[2]}, {Sin[7], Cos[3]}},
{{Sin[4], Cos[2]}, {Sin[5], Cos[1]}, {Sin[6], 1}, {Sin[7], Cos[1]}, {Sin[8], Cos[2]}},
{{Sin[5], Cos[3]}, {Sin[6], Cos[2]}, {Sin[7], Cos[1]}, {Sin[8], 1}, {Sin[9], Cos[1]}},
{{Sin[6], Cos[4]}, {Sin[7], Cos[3]}, {Sin[8], Cos[2]}, {Sin[9], Cos[1]}, {Sin[10], 1}}}
```

```
Length[t1]
```

```
5
```

`Length[list]` returns the number of elements in list.

is not a list of 25 points: `t1` is a list of 5 lists each consisting of 5 points. `t1` has two levels. For example, the third element of the second level is

```
t1[[3]]
```

```
{Sin[4], Cos[2]}, {Sin[5], Cos[1]}, {Sin[6], 1}, {Sin[7], Cos[1]}, {Sin[8], Cos[2]}
```

and the second element of the third level (or the second part of the third part) is

```
t1[[3,2]]
```

```
{Sin[5], Cos[1]}
```

To flatten `t2` to level 1, we use `Flatten`.

```
t2 = Flatten[t1, 1];
```

The resulting list of ordered pairs (in Mathematica, $\{x,y\}$ corresponds to (x,y)) is not displayed because a semicolon is placed at the end of the `Flatten` command. These are plotted with `ListPlot` in Figure 4.6(a). We also illustrate the use of the `PlotStyle`, `PlotRange`, and `AspectRatio` options in the `ListPlot` command.

```
lp1 = ListPlot[t2, PlotStyle -> {PointSize[.05], GrayLevel[.5]},  
PlotRange -> {{-3/2, 3/2}, {-3/2, 3/2}}, AspectRatio -> Automatic];
```

Increasing the number of points further illustrates the use of `Flatten`. Entering

```
t1 = Table[{Sin[x + y], Cos[x - y]}, {x, 1, 125}, {y, 1, 125}];  
Length[t1]  
125
```

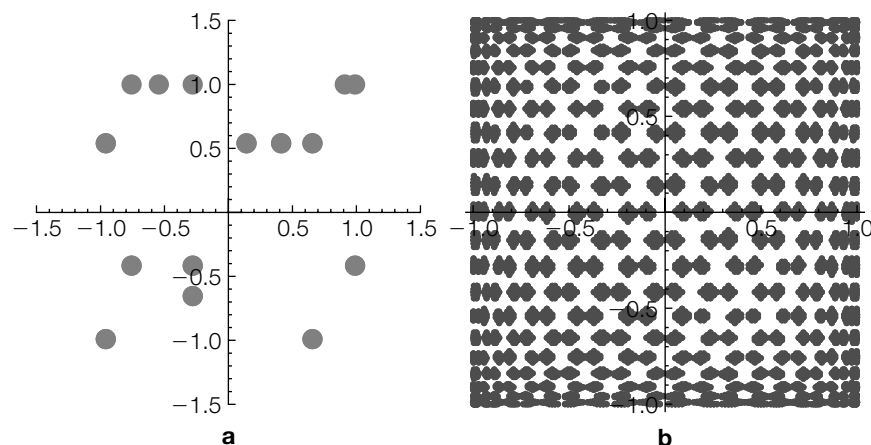


FIGURE 4.6

(a) and (b)

results in a very long nested list. `t1` has 125 elements, each of which has 125 elements.

An abbreviated version is viewed with `Short`.

`Short[list]` yields an abbreviated version of list.

```
Short[t1]
{{{Sin[2], 1}, {Sin[3], Cos[1]}, <<121>>,
{Sin[125], Cos[123]}, {Sin[126], Cos[124]}}, <<124>>}}
```

After using `Flatten`, we see with `Length` and `Short` that `t2` contains 15,625 points,

```
t2 = Flatten[t1, 1];
Length[t2]
15625
Short[t2]
{{Sin[2], 1}, {Sin[3], Cos[1]}, <<15621>>, {Sin[249], Cos[1]}, {Sin[250], 1}}
```

which are plotted with `ListPlot` in Figure 4.6(b).

```
lp2 = ListPlot[t2, AspectRatio -> Automatic];
Show[GraphicsRow[{lp1, lp2}]]
Show[GraphicsRow[{lp1, lp2}]]
```

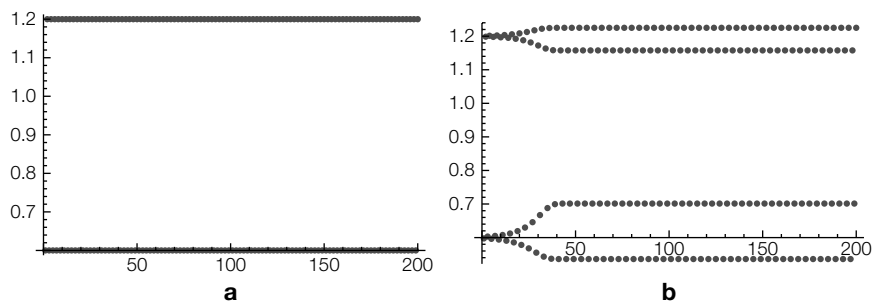
Remark 4.1 Mathematica is very flexible and most calculations can be carried out in more than one way. Depending on how you think, some sequences of calculations may make more sense to you than others, even if they are less efficient than the most efficient way to perform the desired calculations. Often, the difference in time required for Mathematica to perform equivalent—but different—calculations is quite small. For the beginner, we think it is wisest to work with familiar calculations first and then efficiency.

Example 4.1.9 (Dynamical Systems). A sequence of the form $x_{n+1} = f(x_n)$ is called a **dynamical system**.

Sometimes, unusual behavior can be observed when working with dynamical systems. For example, consider the dynamical system with $f(x) = x + 2.5x(1 - x)$ and $x_0 = 1.2$. Note that we define x_n using the form `x[n.] := x[n] = ...` so that Mathematica “remembers” the functional values it computes and thus avoids recomputing functional values previously computed. This is particularly advantageous when we compute the value of x_n for large values of n .

Observe that $x_{n+1} = f(x_n)$ can also be computed with $x_{n+1} = f^n(x_0)$.

```
Clear[f, x]
f[x_] := x + 2.5x(1 - x)
x[n_] := x[n] = f[x[n - 1]]
x[0] = 1.2;
```

**FIGURE 4.7**

(a) A 2-cycle. (b) A 4-cycle

In Figure 4.7(a), we see that the sequence x_n oscillates between the numbers 0.6 and 1.2. We say that the dynamical system has a **2-cycle** because the values of the sequence oscillate between two numbers.

```
tb = Table[x[n], {n, 1, 200}]; ListPlot[tb]
```

In Figure 4.7(b), we see that changing x_0 from 1.2 to 1.201 results in a 4-cycle.

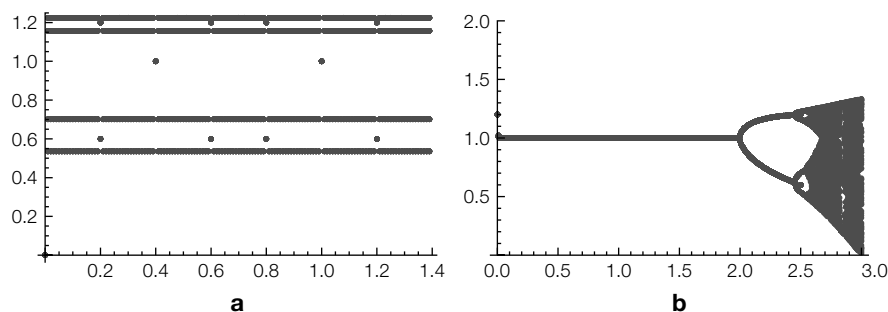
```
Clear[f, x]
f[x_] := x + 2.5x(1 - x)
x[n_] := x[n] = f[x[n - 1]]
x[0] = 1.201;
tb = Table[x[n], {n, 1, 200}];
ListPlot[tb]
```

The calculations indicate that the behavior of the system can change considerably for small changes in x_0 . With the following, we adjust the definition of x so that x depends on $x_0 = c$: Given c , $x_c(0) = c$.

```
Clear[f, x]
f[x_] := x + 2.5x(1 - x)
x[c_][n_] := x[c][n] = f[x[c][n - 1]]/N
x[c_][0] := c/N;
```

In `tb`, we create a list of lists of the form $\{x_c(n) | n = 100, \dots, 150\}$ for 150 equally spaced values of c between 0 and 1.5. Observe that Mathematica issues several error spaced messages. When a Mathematica calculation is larger than the machine's precision, we obtain an **Overflow** warning. In numerical calculations, we interpret **Overflow** to correspond to ∞ .

```
tb = Table[{c, x[c][n]}, {c, 0, 1.5, .01}, {n, 100, 150}];
General::ovfl : Overflow occurred in computation. >>
General::ovfl : Overflow occurred in computation. >>
General::ovfl : Overflow occurred in computation. >>
General::stop : Further output of General::ovfl will be suppressed during this
calculation. >>
```

**FIGURE 4.8**

(a) and (b)

`Short[expr]` prints an abbreviated form of `expr`.

We ignore the error messages and use `Short` to view an abbreviated form of `tb`.

```
Short[tb]
```

```
{{{0., 0.}, {0., 0.}, {0., 0.}, <<45>>, {0., 0.}, {0., 0.}, {0., 0.}, <<149>>, {{{1}}}}
```

We then use `Flatten` to convert `tb` to a list of points that are plotted with `ListPlot` in Figure 4.8(a).

```
tb2 = Flatten[tb, 1];
```

```
f1 = ListPlot[tb2];
```

Another interesting situation occurs if we fix x_0 and let c vary in $f(x) = x + cx(1 - x)$.

With the following, we set $x_0 = 1.2$ and adjust the definition of f so that f depends on c : $f(x) = x + cx(1 - x)$.

```
Clear[f, x]
```

```
f[c_][x_] := x + cx(1 - x)/N
```

```
x[c_][n_] := x[c][n] = f[c][x[c][n - 1]]/N
```

```
x[c_][0] := 1.2/N;
```

In `tb`, we create a list of lists of the form $\{x_c(n) | n = 200, \dots, 300\}$ for 350 equally spaced values of c between 0 and 3.5. As before, Mathematica issues several error messages, which we ignore and which are not displayed here due to length considerations.

```
tb = Table[{c, x[c][n]}, {c, 0, 3.5, .01}, {n, 200, 300}];
```

```
Short[tb]
```

```
{{{0., 1.2}, {0., 1.2}, {0., 1.2}, <<95>>, {0., 1.2}, {0., 1.2}, {0., 1.2}, <<350>>}}
```

`tb` is then converted to a list of points with `Flatten` and the resulting list is plotted in Figure 4.8(b) with `ListPlot`. This plot is called a **bifurcation diagram**.

```
tb2 = Flatten[tb, 1];
```

```
f2 = ListPlot[tb2, PlotRange -> {0, 2}]
```

```
Show[GraphicsRow[{f1, f2}]]
```

A function f is **listable** if $f[\text{list}]$ and $\text{Map}[f, \text{list}]$ return the same results.

As indicated previously, elements of lists can be numbers, ordered pairs, functions, and even other lists. You can also use Mathematica to manipulate lists in numerous ways. Most important, the **Map** function is used to apply a function to a list: $\text{Map}[f, \{x_1, x_2, \dots, x_n\}]$ returns the list $\{f(x_1), f(x_2), \dots, f(x_n)\}$. We discuss other operations that can be performed on lists in the following sections.

Example 4.1.10 (Hermite Polynomials). The **Hermite polynomials**, $H_n(x)$, satisfy the differential equation $y'' - 2xy' + 2ny = 0$ and the orthogonality relation $\int_{-\infty}^{\infty} H_n(x)H_m(x)e^{-x^2} dx = \delta_{mn}2^n n! \sqrt{\pi}$. The Mathematica command **HermiteH** $[n, x]$ yields the Hermite polynomial $H_n(x)$. (a) Create a table of the first five Hermite polynomials. (b) Evaluate each Hermite polynomial if $x = 1$. (c) Compute the derivative of each Hermite polynomial in the table. (d) Compute an antiderivative of each Hermite polynomial in the table. (e) Graph the five Hermite polynomials on the interval $[-1, 1]$. (f) Verify that $H_n(x)$ satisfies $y'' - 2xy' + 2ny = 0$ for $n = 1, 2, \dots, 5$ (' denotes d/dx).

Solution We proceed by using **HermiteH** together with **Table** to define **hermitetable** to be the list consisting of the first five Hermite polynomials.

hermitetable = Table[HermiteH[n, x], {n, 1, 5}]

$\{2x, -2 + 4x^2, -12x + 8x^3, 12 - 48x^2 + 16x^4, 120x - 160x^3 + 32x^5\}$

We then use **ReplaceAll** ($\rightarrow$) to evaluate each member of **hermitetable** if x is replaced by 1.

hermitetable/.x → 1

$\{2, 2, -4, -20, -8\}$

Functions such as **D** and **Integrate** are listable. Thus, each of the following commands differentiates each element of **hermitetable** with respect to x . In the second case, we have used a *pure function*: Given an argument $\#$, **D** $[\#, x] \&$ differentiates $\#$ with respect to x . Use the $\&$ symbol to indicate the end of a pure function.

D[hermitetable, x]

$\{2, 8x, -12 + 24x^2, -96x + 64x^3, 120 - 480x^2 + 160x^4\}$

Map[D[#, x] &, hermitetable]

$\{2, 8x, -12 + 24x^2, -96x + 64x^3, 120 - 480x^2 + 160x^4\}$

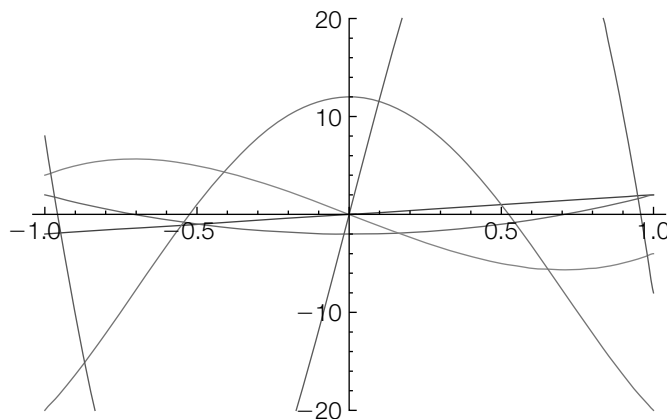
Similarly, we use **Integrate** to antidifferentiate each member of **hermitetable** with respect to x . Remember that Mathematica does not automatically include the $+C$ that we include when we antidifferentiate.

Integrate[hermitetable, x]

$\{x^2, -2x + \frac{4x^3}{3}, -6x^2 + 2x^4, 12x - 16x^3 + \frac{16x^5}{5}, 60x^2 - 40x^4 + \frac{16x^6}{3}\}$

Map[Integrate[#, x] &, hermitetable]

$\{x^2, -2x + \frac{4x^3}{3}, -6x^2 + 2x^4, 12x - 16x^3 + \frac{16x^5}{5}, 60x^2 - 40x^4 + \frac{16x^6}{3}\}$

**FIGURE 4.9**

Graphs of $H_1(x)$, $H_2(x)$, $H_3(x)$, $H_4(x)$, and $H_5(x)$

To graph the list `hermitetable`, we use `Plot` to plot each function in the set `hermitetable` on the interval $[-2, 2]$ in Figure 4.9. In this case, we specify that the displayed y -values correspond to the interval $[-20, 20]$. Because we apply `Tooltip` to the set of functions being plotted, you can identify each curve by moving the cursor and placing it over each curve to see which function is being plotted.

`Plot[Tooltip[hermitetable], {x, -1, 1}, PlotRange → {-20, 20}]`

`hermitetable[[n]]` returns the n th element of `hermitetable`, which corresponds to $H_n(x)$. Thus,

`verifyde =`
`Table[D[hermitetable[[n]], {x, 2}] - 2x D[hermitetable[[n]], x] +`
`2n hermitetable[[n]]//Simplify, {n, 1, 5}]`
`{0, 0, 0, 0, 0}`

computes and simplifies $H_n'' - 2xH_n' + 2nH_n$ for $n = 1, 2, \dots, 5$. We use `Table` and `Integrate` to compute $\int_{-\infty}^{\infty} H_n(x)H_m(x)e^{-x^2} dx$ for $n = 1, 2, \dots, 5$ and $m = 1, 2, \dots, 5$.

`verifyortho =`
`Table[Integrate[hermitetable[[n, 2]]hermitetable[[m, 2]]`
`Exp[-x^2], {x, -Infinity, Infinity}], {n, 1, 5}, {m, 1, 5}]`
`{ {  $\frac{\sqrt{\pi}}{2}, 0, 6\sqrt{\pi}, 0, -120\sqrt{\pi}$  }, { 0, 12\sqrt{\pi}, 0, -144\sqrt{\pi}, 0 },`
`{ 6\sqrt{\pi}, 0, 120\sqrt{\pi}, 0, -2400\sqrt{\pi} }, { 0, -144\sqrt{\pi}, 0, 1728\sqrt{\pi}, 0 },`
`{ -120\sqrt{\pi}, 0, -2400\sqrt{\pi}, 0, 48000\sqrt{\pi} } }`

To view a table in traditional row-and-column form use `TableForm`, as we do here illustrating the use of the `TableHeadings` option.

```
TableForm[verifyortho,
  TableHeadings → {{“m = 1”, “m = 2”, “m = 3”, “m = 4”, “m = 5”},
    {“n = 1”, “n = 2”, “n = 3”, “n = 4”, “n = 5”}}
```

	n = 1	n = 2	n = 3	n = 4	n = 5
m = 1	$\frac{\sqrt{\pi}}{2}$	0	$6\sqrt{\pi}$	0	$-120\sqrt{\pi}$
m = 2	0	$12\sqrt{\pi}$	0	$-144\sqrt{\pi}$	0
m = 3	$6\sqrt{\pi}$	0	$120\sqrt{\pi}$	0	$-2400\sqrt{\pi}$
m = 4	0	$-144\sqrt{\pi}$	0	$1728\sqrt{\pi}$	0
m = 5	$-120\sqrt{\pi}$	0	$-2400\sqrt{\pi}$	0	$48000\sqrt{\pi}$

Be careful when using TableForm: TableForm[table] is no longer a list and cannot be manipulated like a list.

4.2 MANIPULATING LISTS: MORE ON PART AND MAP

Often, Mathematica's output is given to us as a list that we need to use in subsequent calculations. Elements of a list are extracted with Part ([[. . .]]): list[[i]] returns the *i*th element of list, list[[i,j]] (or list[[i]][[j]]) returns the *j*th element of the *i*th element of list, and so on.

Example 4.2.1 Let $f(x) = 3x^4 - 8x^3 - 30x^2 + 72x$. Locate and classify the critical points of $y = f(x)$.

Solution We begin by clearing all prior definitions of f and then defining f . The critical numbers are found by solving the equation $f'(x) = 0$. The resulting list is named critnums.

```
Clear[f]
f[x_] = 3x^4 - 8x^3 - 30x^2 + 72x;
critnums = Solve[f'[x]==0]
{{x → -2}, {x → 1}, {x → 3}}
```

critnums is actually a list of lists. For example, the number -2 is the second part of the first part of the second part of critnums.

```
critnums[[1]]
{x → -2}
critnums[[1, 1]]
x → -2
critnums[[1, 1, 2]]
-2
```

Similarly, the numbers 1 and 3 are extracted with `critnums[[2,1,2]]` and `critnums[[3,1,2]]`, respectively.

```
critnums[[2, 1, 2]]
```

```
critnums[[3, 1, 2]]
```

```
1
```

```
3
```

We locate and classify the points by evaluating $f(x)$ and $f''(x)$ for each of the numbers in `critnums`. `f[x]/.x->a` replaces each occurrence of x in $f(x)$ by a , so entering

```
{{x, f[x], f''[x]}}/.critnums
```

```
{{{-2, -152, 180}, {1, 37, -72}, {3, -27, 120}}}
```

replaces each x in the list $\{x, f(x), f''(x)\}$ by each of the x -values in `critnums`.

By the second derivative test, we conclude that $y = f(x)$ has relative minima at the points $(-2, -152)$ and $(3, -27)$, whereas $f'(x)$ has a relative maximum at $(1, 37)$. In fact, because $\lim_{x \rightarrow \pm\infty} = \infty$, -152 is the absolute minimum value of $f(x)$. These results are confirmed by the graph of $y = f(x)$ in Figure 4.10.

```
Plot[Tooltip[{f[x], f'[x], f''[x]}], {x, -4, 4}]
```

`Map` is a very powerful and useful function: `Map[f,list]` creates a list consisting of elements obtained by evaluating f for each element of `list`, provided that each member of `list` is an element of the domain of f . Note that if f is **listable**, `f[list]` produces the same result as `Map[f,list]`.

When you plot lists of functions and apply `Tooltip` to the list being plotted, you can identify each curve by sliding the cursor over the curve. When the cursor is on a curve, the definition of the curve being plotted is displayed.

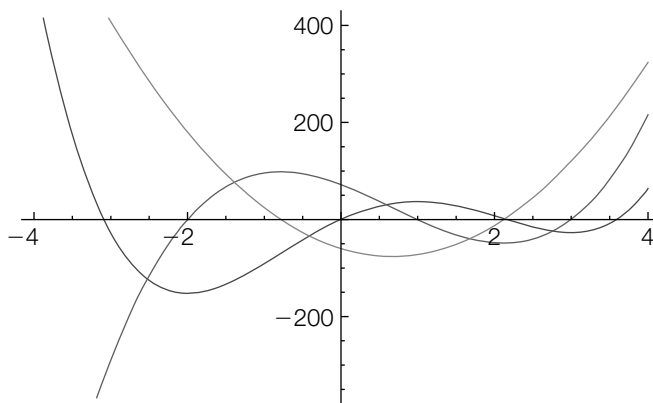


FIGURE 4.10

Graph of $f(x) = 3x^4 - 8x^3 - 30x^2 + 72x$, $f'(x)$, and $f''(x)$

Example 4.2.2 Entering

To determine if f is
listable, enter
`Attributes[f]`.

```
t1 = Table[n, {n, 1, 100}];
t1b = Partition[t1, 10];
TableForm[t1b]
  1  2  3  4  5  6  7  8  9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
```

computes a list of the first 100 integers and names the result `t1`. To see `t1`, we use `Partition` to partition `t1` in 10 element subsets; the results are displayed in a standard row-and-column form with `TableForm`. We then define $f(x) = x^2$ and use `Map` to square each number in `t1`.

```
f[x_] = x^2,
t2 = Map[f, t1];
t2b = Partition[t2, 10];
TableForm[t2b]
  1    4    9   16   25   36   49   64   81   100
121  144  169  196  225  256  289  324  361  400
441  484  529  576  625  676  729  784  841  900
961 1024 1089 1156 1225 1296 1369 1444 1521 1600
1681 1764 1849 1936 2025 2116 2209 2304 2401 2500
2601 2704 2809 2916 3025 3136 3249 3364 3481 3600
3721 3844 3969 4096 4225 4356 4489 4624 4761 4900
5041 5184 5329 5476 5625 5776 5929 6084 6241 6400
6561 6724 6889 7056 7225 7396 7569 7744 7921 8100
8281 8464 8649 8836 9025 9216 9409 9604 9801 10000
```

The same result is accomplished by the pure function that squares its argument. Note how `#` denotes the argument of the pure function; the `&` symbol marks the end of the pure function.

```

t3 = Map[#^2&, t1];
t3b = Partition[t3, 10];
TableForm[t3b]

```

1	4	9	16	25	36	49	64	81	100
121	144	169	196	225	256	289	324	361	400
441	484	529	576	625	676	729	784	841	900
961	1024	1089	1156	1225	1296	1369	1444	1521	1600
1681	1764	1849	1936	2025	2116	2209	2304	2401	2500
2601	2704	2809	2916	3025	3136	3249	3364	3481	3600
3721	3844	3969	4096	4225	4356	4489	4624	4761	4900
5041	5184	5329	5476	5625	5776	5929	6084	6241	6400
6561	6724	6889	7056	7225	7396	7569	7744	7921	8100
8281	8464	8649	8836	9025	9216	9409	9604	9801	10000

On the other hand, entering

```

t1 = Table[{a, b}, {a, 1, 5}, {b, 1, 5}];
Short[t1]

```

```

{{{1, 1}, {1, 2}, {1, 3}, {1, 4}, {1, 5}}, <<4>>}}

```

is a list (of length 5) of lists (each of length 5). Use `Flatten` to obtain a list of 25 points, which we name `t2`.

```

t2 = Flatten[t1, 1];
Short[t2]

```

```

{{1, 1}, {1, 2}, {1, 3}, {1, 4}, <<17>>, {5, 2}, {5, 3}, {5, 4}, {5, 5}}

```

We then use `Map` to apply f to `t2`.

```

f[{x_, y_}] = {{x, y}, x^2 + y^2};
t3 = Map[f, t2];
Short[t3]

```

```

{{{1, 1}, 2}, {{1, 2}, 5}, {{1, 3}, 10}, <<20>>, {{5, 4}, 41}, {{5, 5}, 50}}

```

We accomplish the same result with a pure function. Observe how `#[[1]]` and `#[[2]]` are used to represent the first and second arguments: Given a list of length 2, the pure function returns the list of ordered pairs consisting of the first element of the list, the second element of the list (as an ordered pair), and the sum of the squares of the first and second elements (of the first ordered pair).

```

t3b = Map[{{#[[1]], #[[2]]}, #[[1]]^2 + #[[2]]^2}&, t2];
Short[t3b]

```

```

{{{1, 1}, 2}, {{1, 2}, 5}, {{1, 3}, 10}, <<20>>, {{5, 4}, 41}, {{5, 5}, 50}}

```

Example 4.2.3 Make a table of the values of the trigonometric functions $y = \sin x$, $y = \cos x$, and $y = \tan x$ for the principal angles.

Solution We first construct a list of the principal angles, which is accomplished by defining **t1** to be the list consisting of $n\pi/4$ for $n = 0, 1, \dots, 8$ and **t2** to be the list consisting of $n\pi/6$ for $n = 0, 1, \dots, 12$. The principal angles are obtained by taking the union of **t1** and **t2**. **Union[t1,t2]** joins the lists **t1** and **t2**, removes repeated elements, and sorts the results. If we did not wish to remove repeated elements and sort the result, the command **Join[t1,t2]** concatenates the lists **t1** and **t2**.

```
t1 = Table [ $\frac{n\pi}{4}$ , {n, 0, 8}];
t2 = Table [ $\frac{n\pi}{6}$ , {n, 0, 12}];
prinangles = Union[t1, t2]
{0,  $\frac{\pi}{6}$ ,  $\frac{\pi}{4}$ ,  $\frac{\pi}{3}$ ,  $\frac{\pi}{2}$ ,  $\frac{2\pi}{3}$ ,  $\frac{3\pi}{4}$ ,  $\frac{5\pi}{6}$ ,  $\pi$ ,  $\frac{7\pi}{6}$ ,  $\frac{5\pi}{4}$ ,  $\frac{4\pi}{3}$ ,  $\frac{3\pi}{2}$ ,  $\frac{5\pi}{3}$ ,  $\frac{7\pi}{4}$ ,  $\frac{11\pi}{6}$ ,  $2\pi$ }
```

The **BasicMathInput** palette:



Remember that the result of using **TableForm** is not a list, so it cannot be manipulated like lists.

We can also use the symbol $\cup$, which is obtained by clicking on the $\cup$ button on the **BasicMathInput** palette to represent **Union**.

```
prinangles = t1  $\cup$  t2
{0,  $\frac{\pi}{6}$ ,  $\frac{\pi}{4}$ ,  $\frac{\pi}{3}$ ,  $\frac{\pi}{2}$ ,  $\frac{2\pi}{3}$ ,  $\frac{3\pi}{4}$ ,  $\frac{5\pi}{6}$ ,  $\pi$ ,  $\frac{7\pi}{6}$ ,  $\frac{5\pi}{4}$ ,  $\frac{4\pi}{3}$ ,  $\frac{3\pi}{2}$ ,  $\frac{5\pi}{3}$ ,  $\frac{7\pi}{4}$ ,  $\frac{11\pi}{6}$ ,  $2\pi$ }
```

Next, we define $f(x)$ to be the function that returns the ordered quadruple $(x, \sin x, \cos x, \tan x)$ and compute the value of $f(x)$ for each number in **prinangles** with **Map** naming the resulting table **prinvalues**. **prinvalues** is not displayed because a semicolon is included at the end of the command.

```
Clear[f]
f[x_] = {x, Sin[x], Cos[x], Tan[x]};
prinvalues = Map[f, prinangles];
```

Finally, we use **TableForm** illustrating the use of the **TableHeadings** option to display **prinvalues** in row-and-column form; the columns are labeled x , $\sin x$, $\cos x$, and $\tan x$.

```
TableForm[prinvalues,
  TableHeadings  $\rightarrow$  {None, {"x", "sin(x)", "cos(x)", "tan(x)"}}]
```

x	sin(x)	cos(x)	tan(x)
0	0	1	0
$\frac{\pi}{6}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{3}}$
$\frac{\pi}{4}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	1
$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\sqrt{3}$
$\frac{\pi}{2}$	1	0	ComplexInfinity
$\frac{2\pi}{3}$	$\frac{\sqrt{3}}{2}$	$-\frac{1}{2}$	$-\sqrt{3}$
$\frac{3\pi}{4}$	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	-1
$\frac{5\pi}{6}$	$\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$	$-\frac{1}{\sqrt{3}}$
π	0	-1	0

$\frac{7\pi}{6}$	$-\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{3}}$
$\frac{5\pi}{4}$	$-\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	1
$\frac{4\pi}{3}$	$-\frac{\sqrt{3}}{2}$	$-\frac{1}{2}$	$\sqrt{3}$
$\frac{3\pi}{2}$	-1	0	ComplexInfinity
$\frac{5\pi}{3}$	$-\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$-\sqrt{3}$
$\frac{7\pi}{4}$	$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	-1
$\frac{11\pi}{6}$	$-\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$-\frac{1}{\sqrt{3}}$
2π	0	1	0

In the table, note that $y = \tan x$ is undefined at odd multiples of $\pi/2$ and Mathematica appropriately returns `ComplexInfinity` at those values of x for which $y = \tan x$ is undefined.

Remark 4.2

object=name assigns the object object the name name.

We can use `Map` on any list, including lists of functions and/or other lists.

The result of using `TableForm` is not a list (or table) and calculations on it using commands such as `Map` cannot be performed. `TableForm` helps you see results in a more readable format. To avoid confusion, do not assign the results of using `TableForm` any name: Adopting this convention avoids any possible attempted manipulation of `TableForm` objects.

Lists of functions are graphed with `Plot`: `Plot[listoffunctions, {x,a,b}]` graphs the list of functions of x , `listoffunctions`, for $a \leq x \leq b$. If the command is entered as `Plot[Tooltip[listoffunctions], {x,a,b}]`, you can identify the curves in the plot by moving the cursor over the curves in the graphic.

Example 4.2.4

(Bessel Functions). The **Bessel functions of the first kind**, $J_n(x)$, are non-singular solutions of $x^2 y'' + xy' + (x^2 - n^2)y = 0$. `BesselJ[n,x]` returns $J_n(x)$. Graph $J_n(x)$ for $n = 0, 1, 2, \dots, 8$.

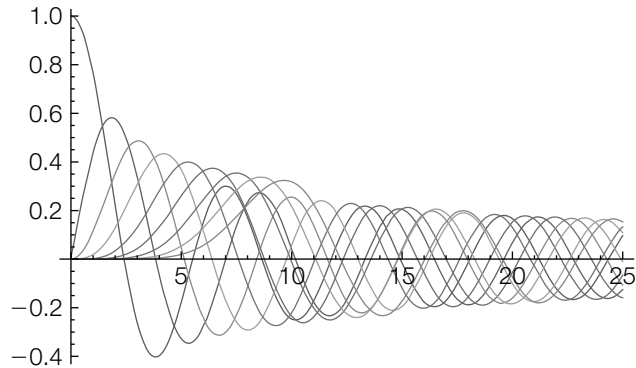
Solution In `t1`, we use `Table` and `BesselJ` to create a list of $J_n(x)$ for $n = 0, 1, 2, \dots, 8$.

```
t1 = Table[BesselJ[n, x], {n, 0, 8}];
```

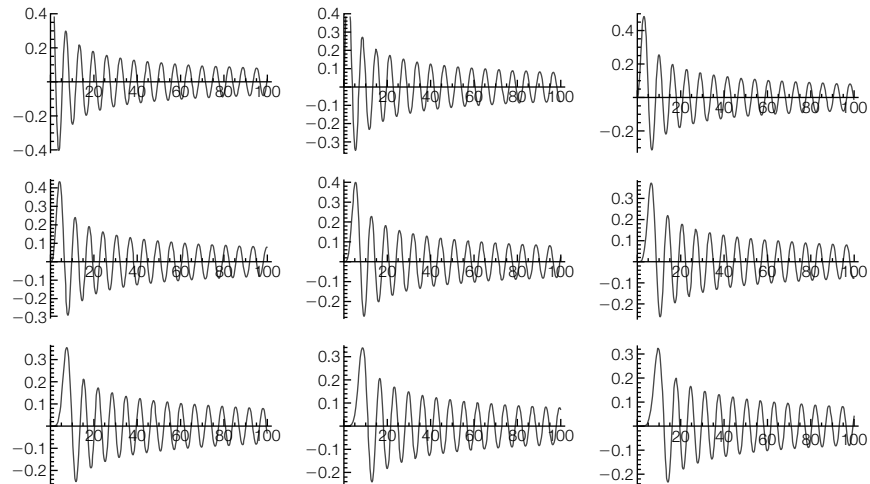
We then use `Plot` to graph each function in `t1` in Figure 4.11. You can identify each curve by sliding the cursor over each.

```
Plot[Tooltip[t1], {x, 0, 25}]
```

A different effect is achieved by graphing each function separately. To do so, we define the function `pfunc`. Given a function of x , `f`, `pfunc[f]` plots the function for $0 \leq x \leq 100$. The resulting graphic is not displayed because the option `DisplayFunction->Identity` is included in the `Plot` command. We then use `Map` to apply `pfunc` to each element of `t1`. The result is a list of nine graphics objects, which we name `t2`. A good way to display nine graphics is as a 3×3 array, so we use `Partition` to convert `t2` from a list of length 9 to a list of lists, each with length

**FIGURE 4.11**

Graphs of $J_n(x)$ for $n = 0, 1, 2, \dots, 8$

**FIGURE 4.12**

In the first row, from left to right, graphs of $J_0(x)$, $J_1(x)$, and $J_2(x)$; in the second row, from left to right, graphs of $J_3(x)$, $J_4(x)$, and $J_5(x)$; in the third row, from left to right, graphs of $J_6(x)$, $J_7(x)$, and $J_8(x)$

Think of Flatten and Partition as inverse functions.

3—a 3×3 array. Partition[list,n] returns a list of lists obtained by partitioning list into n -element subsets.

```
pfunc[f_] := Plot[f, {x, 0, 100}];
t2 = Map[pfunc, t1];
t3 = Partition[t2, 3];
```

Instead of defining pfunc, you can use a pure function. The following accomplishes the same result. We display t3 using Show together with GraphicsGrid in Figure 4.12.

```

t2 = (Plot[#1, {x, 0, 100}, DisplayFunction → Identity]&)/@t1;
t3 = Partition[t2, 3];
Show[GraphicsGrid[t3]]

```

Example 4.2.5 (Dynamical Systems). Let $f_c(x) = x^2 + c$ and consider the dynamical system given by $x_0 = 0$ and $x_{n+1} = f_c(x_n)$. Generate a bifurcation diagram of f_c .

Solution First, recall that `Nest[f,x,n]` computes the repeated composition $f^n(x)$. Then, in terms of a composition,

Compare the approach used here with the approach used in Example 4.1.9.

$$x_{n+1} = f_c(x_n) = f_c^n(0).$$

We will compute $f_c^n(0)$ for various values of c and “large” values of n so we begin by defining `cvals` to be a list of 300 equally spaced values of c between -2.5 and 1 .

```
cvals = Table[c, {c, -2.5, 1., 3.5/299}];
```

We then define $f_c(x) = x^2 + c$. For a given value of c , `f[c]` is a function of one variable, x , whereas the form `f[c.][x.]:=...` results in a function of two variables that we think of as an indexed function that might be represented using traditional mathematical notation as $f_c(x)$.

```

Clear[f]
f[c.][x.]:= x^2 + c

```

To iterate f_c for various values of c , we define `h`. For a given value of c , `h(c)` returns the list of points $\{(c, f_c^{100}(0)), (c, f_c^{101}(0)), \dots, (c, f_c^{200}(0))\}$.

```
h[c_] := {Table[{c, Nest[f[c], 0, n]}, {n, 100, 200}]}
```

We then use `Map` to apply `h` to the list `cvals`. Observe that Mathematica generates several error messages when numerical precision is exceeded. We choose to disregard the error messages.

```
t1 = Map[h, cvals];
```

`t1` is a list (of length 300) of lists (each of length 101). To obtain a list of points (or, lists of length 2), we use `Flatten`. The resulting set of points is plotted with `ListPlot` in Figure 4.13. Observe that Mathematica again displays several error messages, which are not displayed here for length considerations, that we ignore: Mathematica only plots the points with real coordinates and ignores those containing `Overflow[]`.

```

t2 = Flatten[t1, 2];
ListPlot[t2, AxesLabel → {"c", "x_c(n), n = 100 . . 200"}]

```

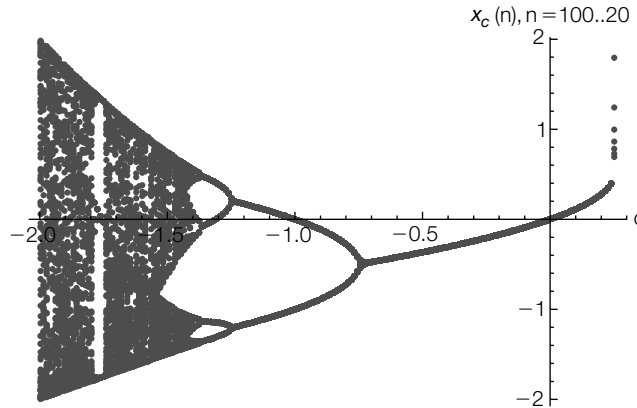


FIGURE 4.13

Bifurcation diagram of f_c

4.2.1 More on Graphing Lists: Graphing Lists of Points Using Graphics Primitives

Include the `PlotJoined->True` option in a `ListPlot` command to connect successive points with line segments.

Using *graphics primitives* such as `Point` and `Line` gives you even more flexibility. `Point[{x,y}]` represents a point at (x,y) .

`Line[{{x1,y1},{x2,y2},...,{xn,yn}}]`

represents a sequence of points $(x_1,y_1), (x_2,y_2), \dots, (x_n,y_n)$ connected with line segments. A graphics primitive is declared to be a graphics object with `Graphics: Show[Graphics[Point[x,y]]` displaying the point (x,y) . The advantage of using primitives is that each primitive is affected by the options that directly precede it.

Example 4.2.6 Table 4.1 shows the percentage of the U.S. labor force that belonged to unions during certain years. Graph the data represented in the table.

Solution We begin by entering the data represented in the table as `dataunion`:

```
dataunion = {{30, 11.6}, {35, 13.2}, {40, 26.9}, {45, 35.5},
             {50, 31.5}, {55, 33.2}, {60, 31.4}, {65, 28.4}, {70, 27.3},
             {75, 25.5}, {80, 21.9}, {85, 18.0}, {90, 16.1}};
```

the x -coordinate of each point corresponds to the year, where x is the number of years past 1900, and the y -coordinate of each point corresponds to the percentage of the U.S. labor force that belonged to unions in the given year. We then use `ListPlot` to graph the set of points represented in `dataunion` in `lp1`, `lp2` (illustrating

Table 4.1 Union membership as a percentage of the labor force

Year	Percent
1930	11.6
1935	13.2
1940	26.9
1945	35.5
1950	31.5
1955	33.2
1960	31.4
1965	28.4
1970	27.3
1975	25.5
1980	21.9
1985	18.0
1990	16.1

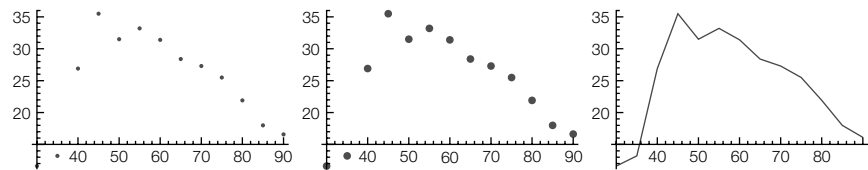


FIGURE 4.14

Union membership as a percentage of the labor force

the `PlotStyle` option), and `lp3` (illustrating the `PlotJoined` option). All three plots are displayed side-by-side in Figure 4.14 using `Show` together with `GraphicsRow`.

```
lp1 = ListPlot[dataunion];
lp2 = ListPlot[dataunion, PlotStyle -> PointSize[0.03]];
lp3 = ListPlot[dataunion, Joined -> True];
Show[GraphicsRow[{lp1, lp2, lp3}]]
```

An alternative to using `ListPlot` is to use `Show`, `Graphics`, and `Point` to view the data represented in `dataunion`. In the following command we use `Map` to apply the function `Point` to each pair of data in `dataunion`. The result is not a graphics object and cannot be displayed with `Show`.

```
datapts1 = Map[Point, dataunion];
Short[datapts1]
{Point[{30, 11.6}], Point[{35, 13.2}], <(10)>, Point[{90, 16.1}]}
```

Next, we use `Show` and `Graphics` to declare the set of points `Map[Point, dataunion]` as graphics objects and name the resulting graphics object `dp1`. The

image is not displayed because a semicolon is included at the end of the command. The `PointSize[.03]` command specifies that the points be displayed as filled circles of radius 0.03% of the displayed graphics object.

```
dp1 = Show[Graphics[{PointSize[0.03], datapts1},  
Axes → Automatic];
```

The collection of all commands contained within a `Graphics` command is contained in braces `{...}`. Each graphics primitive is affected by the options such as `PointSize`, `GrayLevel` (or `RGBColor`) directly preceding it. Thus,

```
datapts2 = ({GrayLevel[RandomReal[], Point[#1]]&)/@dataunion;  
Short[datapts2]  
{ {GrayLevel[0.827228], Point[{30, 11.6}]}, {{11}}, {{1}}}  
  
dp2 = Show[Graphics[{PointSize[0.03], datapts2},  
Axes → Automatic];
```

displays the points in `dataunion` in various shades of gray in a graphic named `dp2`, and

```
datapts3 = ({PointSize[RandomReal[{"0.008", "0.1"}],  
GrayLevel[RandomReal[]], Point[#1]]&)/@dataunion;  
dp3 = Show[Graphics[{datapts3}, Axes → Automatic];
```

shows the points in `dataunion` in various sizes and in various shades of gray in a graphic named `dp3`. We connect successive points with line segments

```
connectpts = Graphics[Line[dataunion]];  
dp4 = Show[connectpts, dp3, Axes → Automatic];
```

and show all four plots in Figure 4.15 using `Show` and `GraphicsGrid`.

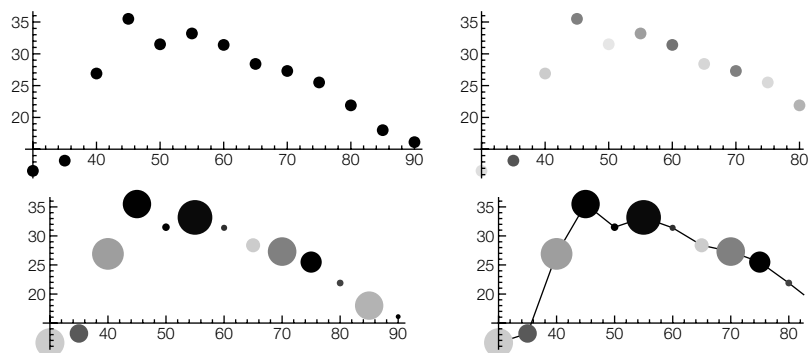
```
Show[GraphicsGrid[{{dp1, dp2}, {dp3, dp4}}]]
```

With the speed of today's computers and the power of Mathematica, it is relatively easy to carry out many calculations that required supercomputers and sophisticated programming experience just a few years ago.

Example 4.2.7 (Julia Sets). Plot Julia sets for $f(z) = \lambda \cos z$ if $\lambda = .66i$ and $\lambda = .665i$.

Solution The sets are visualized by plotting the points (a, b) for which $|f^n(a + bi)|$ is not large in magnitude so we begin by forming our complex grid. Using `Table` and `Flatten`, we define `complexpts` to be a list of 62,500 points of the form $a + bi$ for 250 equally spaced real values of a between 0 and 8 and 300 equally spaced real values of b between -4 and 4 and then $f(z) = .66i \cos z$.

```
complexpts =  
Flatten[Table[a + bi, {a, 0., 8., 8/249}, {b, -4., 4., 6/249}], 1];
```

**FIGURE 4.15**

Union membership as a percentage of the labor force

```
Clear[f]
f[z_] = .66I Cos[z]
0.66I Cos[z]
```

For a given value of $c = a + bi$, $b(c)$ returns the ordered triple consisting of the real part of c , the imaginary part of c , and the value of $f^{200}(c)$.

```
h[c_] := {Re[c], Im[c], Nest[f, c, 200]}
```

We then use `Map` to apply b to `complexpts`. Observe that Mathematica generates several error messages. When machine precision is exceeded, we obtain an `Overflow[]` error message; numerical result smaller than machine precision results in an `Underflow[]` error message. Error messages can be machine specific, so if you do not get any, do not worry. For length considerations, we do not show any that we obtained here.

```
t1 = Map[h, complexpts]//Chop;
```

We use the error messages to our advantage. In `t2`, we select those elements of `t1` for which the third coordinate is *not* `Indeterminate`, which corresponds to the ordered triples $(a, b, f^n(a + bi))$ for which $|f^n(a + bi)|$ is *not* large in magnitude, whereas in `t2b`, we select those elements of `t1` for which the third coordinate is `Indeterminate`, which corresponds to the ordered triples $(a, b, f^n(a + bi))$ for which $|f^n(a + bi)|$ is large in magnitude.

```
t2 = Select[t1, Not[#[[3]] === Indeterminate]&];
t2b = Select[t1, #[[3]] === Indeterminate&];
pt[{x_, y_, z_}] := {x, y}
t3 = Map[pt, t2];
t3b = Map[pt, t2b];
```

which are then graphed with `ListPlot` and shown side-by-side in Figure 4.16 using `Show` and `GraphicsRow`. As expected, the images are inversions of each other.

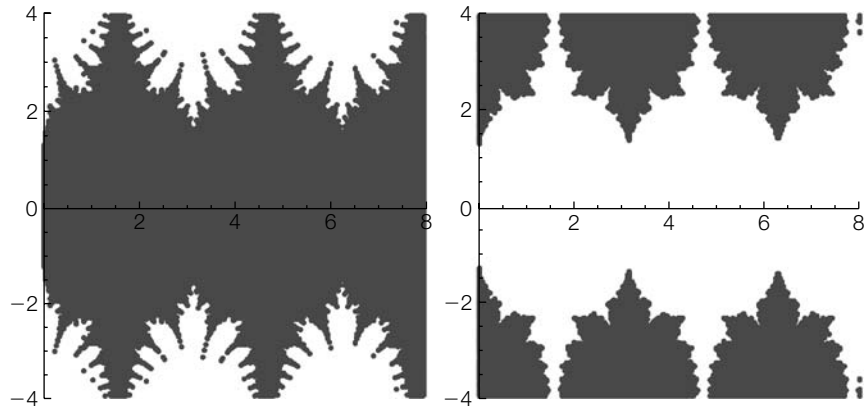


FIGURE 4.16

Julia set for $0.66i \cos z$

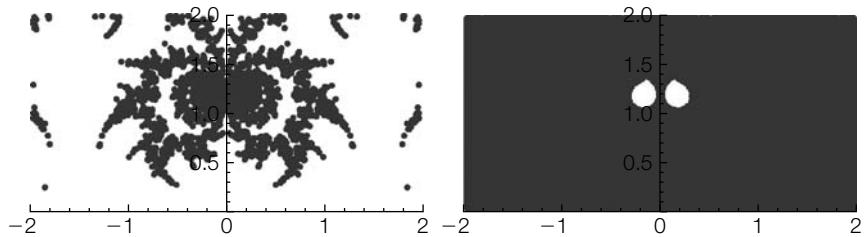


FIGURE 4.17

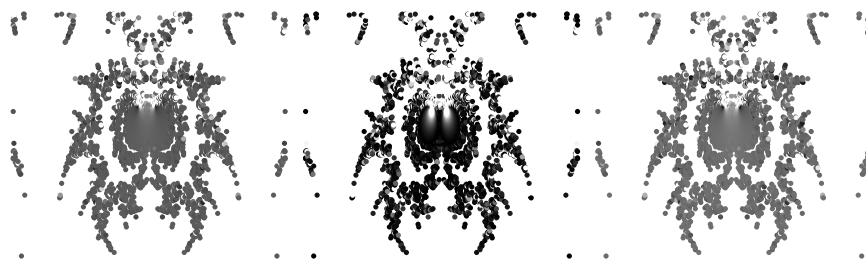
Julia set for $0.665i \cos z$

```
lp1 = ListPlot[t3, PlotRange → {{0, 8}, {-4, 4}},
  AspectRatio → Automatic, DisplayFunction → Identity];
lp2 = ListPlot[t3b, PlotRange → {{0, 8}, {-4, 4}},
  AspectRatio → Automatic, DisplayFunction → Identity];
Show[GraphicsRow[{lp1, lp2}]]
```

We encountered similar error messages as before but we have not included them due to length considerations.

Changing λ from $0.66i$ to $0.665i$ results in a surprising difference in the plots. We proceed as before but increase the number of sample points to 120,000. See Figure 4.17.

```
complexpts = Flatten[Table[a + bi, {a, -2., 2., 4/399}, {b, 0., 2., 2/299}], 1]
Clear[f];
f[z_] = .665 I Cos[z]
h[c_] := {Re[c], Im[c], Nest[f, c, 200]}
t1 = Map[h, complexpts] // Chop;
t2 = Select[t1, Not[#[[3]] === Indeterminate]&];
t2 = Select[t2, Not[#[[3]] === Overflow[]&];
```

**FIGURE 4.18**Shaded Julia sets for $0.665i \cos z$

```

t2b = Select[t1, #[[3]] === Indeterminate&];
pt[{x_, y_, z_}] := {x, y}

t3 = Map[pt, t2];
t3b = Map[pt, t2b];

lp1 = ListPlot[t3, PlotRange → {{-2, 2}, {0, 2}}, AspectRatio → Automatic,
  DisplayFunction → Identity];
lp2 = ListPlot[t3b, PlotRange → {{-2, 2}, {0, 2}}, AspectRatio → Automatic,
  DisplayFunction → Identity];
Show[GraphicsRow[{lp1, lp2}]]

```

To see detail, we take advantage of pure functions, `Map`, and graphics primitives in three different ways. In Figure 4.18, the shading of the point (a, b) is assigned according to the distance of $f^{200}(a + bi)$ from the origin. The color black indicates a distance of zero from the origin; as the distance increases, the shading of the point becomes lighter.

```

t2p = Map[{#[[1]], #[[2]], Min[Abs#[[3]], 3]}&, t2];
t2p2 = Map[{GrayLevel[#[[3]]/3], Point[{#[[1]], #[[2]]]}]}&,
  t2p];
jp1 = Show[Graphics[t2p2], PlotRange → {{-2, 2}, {0, 2}},
  AspectRatio → 1];

t2p = Map[{#[[1]], #[[2]], Min[Abs[Re#[[3]]], .25]}&, t2];
t2p2 = Map[{GrayLevel[#[[3]] /.25], Point[{#[[1]], #[[2]]]}]}&,
  t2p];
jp2 = Show[Graphics[t2p2], PlotRange → {{-2, 2}, {0, 2}}, AspectRatio → 1];

t2p = Map[{#[[1]], #[[2]], Min[Abs[Im#[[3]]], 2.5]}&, t2];
t2p2 = Map[{GrayLevel[#[[3]]/2.5], Point[{#[[1]], #[[2]]]}]}&,
  t2p];
jp3 = Show[Graphics[t2p2], PlotRange → {{-2, 2}, {0, 2}}, AspectRatio → 1];
Show[GraphicsRow[{jp1, jp2, jp3}]]

```

4.2.2 Miscellaneous List Operations

Other List Operations

Some other Mathematica commands used with lists include

1. Append[list,element], which appends element to list;
2. AppendTo[list,element], which appends element to list and names the result list;
3. Drop[list,n], which returns the list obtained by dropping the first n elements from list;
4. Drop[list,-n], which returns the list obtained by dropping the last n elements of list;
5. Drop[list,{n,m}], which returns the list obtained by dropping the n th through m th elements of list;
6. Drop[list,{n}], which returns the list obtained by dropping the n th element of list;
7. Prepend[list,element], which prepends element to list; and
8. PrependTo[list,element], which prepends element to list and names the result list.

Alternative Way to Evaluate Lists by Functions

Abbreviations of several of the commands discussed in this section are summarized in the following table:

/@/@ Apply	// (function application)	{...}
@ Map	[[...]] Part	

4.3 OTHER APPLICATIONS

We now present several other applications that we find interesting and that require the manipulation of lists. The examples also illustrate (and combine) many of the techniques that were demonstrated in the previous chapters.

4.3.1 Approximating Lists with Functions

Another interesting application of lists is that of curve fitting. The commands

1. Fit[data,functionset,variables] fits the list of data points data using the functions in functionset by the method of least squares. The functions in functionset are functions of the variables listed in variables; and

2. `InterpolatingPolynomial[data,x]` fits the list of n data points `data` with an $n - 1$ degree polynomial in the variable x .

Example 4.3.1 Define `datalist` to be the list of numbers consisting of 1.14479, 1.5767, 2.68572, 2.5199, 3.58019, 3.84176, 4.09957, 5.09166, 5.98085, 6.49449, and 6.12113. (a) Find a quadratic approximation of the points in `datalist`. (b) Find a fourth-degree polynomial approximation of the points in `datalist`.

Solution The approximating function obtained via the least squares method with `Fit` is plotted along with the data points in Figure 4.19. Notice that many of the data points are not very close to the approximating function. A better approximation is obtained using a polynomial of higher degree (4).

```
Clear[datalist]
datalist = {1.14479, 1.5767, 2.68572, 2.5199, 3.58019, 3.84176,
           00094.09957, 5.09166, 5.98085, 6.49449, 6.12113};
p1 = ListPlot[datalist];
Clear[y]
y[x_] = Fit[datalist, {1, x, x^2}, x]
0.508266 + 0.608688x - 0.00519281x^2
p2 = Plot[y[x], {x, -1, 11}];
pa = Show[p1, p2];
Clear[y]
y[x_] = Fit[datalist, {1, x, x^2, x^3, x^4}, x]
-0.54133 + 2.02744x - 0.532282x^2 + 0.0709201x^3 - 0.00310985x^4
p3 = Plot[y[x], {x, -1, 11}];
pb = Show[p1, p3];
Show[GraphicsRow[{pa, pb}]]
```

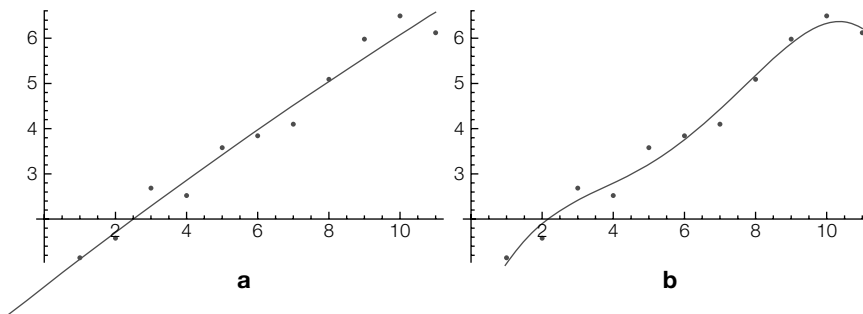


FIGURE 4.19

(a) The graph of a quadratic fit shown with the data points. (b) The graph of a quartic fit shown with the data points

Table 4.2 Petroleum products imported to the United States for certain years

Year	Percent	Year	Percent
1973	34.8105	1983	28.3107
1974	35.381	1984	29.9822
1975	35.8167	1985	27.2542
1976	40.6048	1986	33.407
1977	47.0132	1987	35.4875
1978	42.4577	1988	38.1126
1979	43.1319	1989	41.57
1980	37.3182	1990	42.1533
1981	33.6343	1991	39.5108
1982	28.0988		

Remember that when a semicolon is placed at the end of the command, the resulting output is *not* displayed by Mathematica.

To check its accuracy, the second approximation is graphed simultaneously with the data points in Figure 4.19(b).

Next, consider a list of data points made up of ordered pairs.

Example 4.3.2

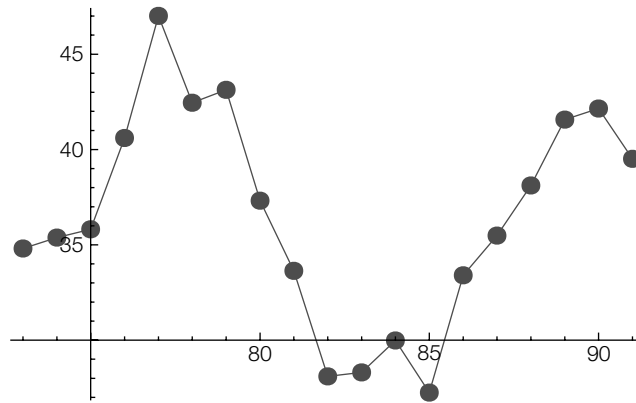
Table 4.2 shows the average percentage of petroleum products imported to the United States for certain years. (a) Graph the points corresponding to the data in the table and connect the consecutive points with line segments. (b) Use `InterpolatingPolynomial` to find a function that approximates the data in the table. (c) Find a fourth-degree polynomial approximation of the data in the table. (d) Find a trigonometric approximation of the data in the table.

Solution

We begin by defining `data` to be the set of ordered pairs represented in the table: The x -coordinate of each point represents the number of years past 1900, and the y -coordinate represents the percentage of petroleum products imported to the United States.

```
data = {{73., 34.8105}, {74., 35.381}, {75., 35.8167},
       {76., 40.6048}, {77., 47.0132}, {78., 42.4577},
       {79., 43.1319}, {80., 37.3182}, {81., 33.6343},
       {82., 28.0988}, {83., 28.3107}, {84., 29.9822},
       {85., 27.2542}, {86., 33.407}, {87., 35.4875},
       {88., 38.1126}, {89., 41.57}, {90., 42.1533}, {91., 39.5108}};
```

We use `ListPlot` to graph the ordered pairs in `data`. Note that because the option `PlotStyle->PointSize[0.03]` is included within the `ListPlot` command, the points are larger than they would normally be. We also use `ListPlot` with the option `PlotJoined->True` to graph the set of points `data` and connect consecutive points with line segments. Then we use `Show` to display `lp1` and `lp2` together in

**FIGURE 4.20**

The points in Table 4.2 connected by line segments

Figure 4.20. Note that in the result, the points are easy to distinguish because of their larger size.

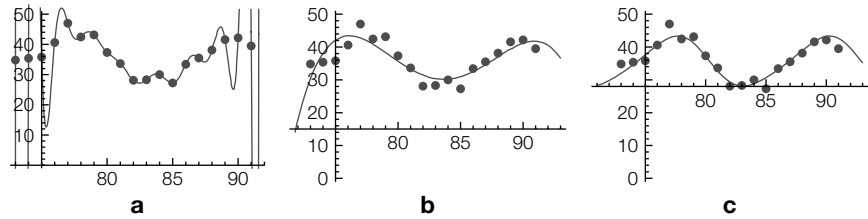
```
lp1 = ListPlot[data, PlotStyle → PointSize[0.03]];
lp2 = ListPlot[data, Joined → True];
Show[lp1, lp2]
```

Next, we use `InterpolatingPolynomial` to find a polynomial approximation, p , of the data in the table. Note that the result is lengthy, so `Short` is used to display an abbreviated form of p . We then graph p and show the graph of p along with the data in the table for the years corresponding to 1971 to 1993 in Figure 4.21(a). Although the interpolating polynomial agrees with the data exactly, the interpolating polynomial oscillates wildly.

```
p = InterpolatingPolynomial[data, x];
Short[p, 3]
39.5108 + (0.261128 + (0.111875 + ((1))) (-82. + x)) (-73. + x) (-91. + x)
plotp = Plot[p, {x, 71, 93}];
pa = Show[plotp, lp1, PlotRange → {0, 50}];
```

To find a polynomial that approximates the data but does not oscillate wildly, we use `Fit`. Again, we graph the fit and display the graph of the fit and the data simultaneously. In this case, the fit does not identically agree with the data but does not oscillate wildly as illustrated in Figure 4.21(b).

```
Clear[p]
p = Fit[data, {1, x, x^2, x^3, x^4}, x]
-198884. + 9597.83x - 173.196x^2 + 1.38539x^3 - 0.00414481x^4
plotp = Plot[p, {x, 71, 93}];
pb = Show[plotp, lp1, PlotRange → {0, 50}]
```

**FIGURE 4.21**

(a) Although interpolating polynomials agree with the data exactly, they may have extreme oscillations, even for relatively small data sets. (b) Although the fit does not agree with the data exactly, the oscillations seen in (a) do not occur. (c) You can use **Fit** to approximate data by a variety of functions

In addition to curve fitting with polynomials, Mathematica can also fit the data with trigonometric functions. In this case, we use **Fit** to find an approximation of the data of the form $p = c_1 + c_2 \sin x + c_3 \sin(x/2) + c_4 \cos x + c_5 \cos(x/2)$. As in the previous two cases, we graph the fit and display the graph of the fit and the data simultaneously; the results are shown in Figure 4.21(c).

See texts such as Abell, Braselton, and Rafter's *Statistics with Mathematica* [3] for a more sophisticated discussion of curve fitting and related statistical applications.

Clear[p]

p = Fit[data, {1, Sin[x], Sin[x/2], Cos[x], Cos[x/2]}, x]

$35.4237 + 4.25768 \cos[\frac{x}{2}] - 0.941862 \cos[x] + 6.06609 \sin[\frac{x}{2}] + 0.0272062 \sin[x]$

plotp = Plot[p, {x, 71, 93}];

pc = Show[plotp, lp1, PlotRange → {0, 50}];

Show[GraphicsRow[{pa, pb, pc}]]

4.3.2 Introduction to Fourier Series

Many problems in applied mathematics are solved through the use of Fourier series. Mathematica assists in the computation of these series in several ways. Suppose that $y = f(x)$ is defined on $-p < x < p$. Then the Fourier series for $f(x)$ is

$$\frac{1}{2}a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{p} + b_n \sin \frac{n\pi x}{p} \right) \quad (4.1)$$

where

$$\begin{aligned} a_0 &= \frac{1}{p} \int_{-p}^p f(x) dx \\ a_n &= \frac{1}{p} \int_{-p}^p f(x) \cos \frac{n\pi x}{p} dx \quad n = 1, 2, \dots \\ b_n &= \frac{1}{p} \int_{-p}^p f(x) \sin \frac{n\pi x}{p} dx \quad n = 1, 2, \dots \end{aligned} \quad (4.2)$$

The k th term of the Fourier series (4.1) is

$$a_n \cos \frac{n\pi x}{p} + b_n \sin \frac{n\pi x}{p}. \quad (4.3)$$

The k th partial sum of the Fourier series (4.1) is

$$\frac{1}{2}a_0 + \sum_{n=1}^k \left(a_n \cos \frac{n\pi x}{p} + b_n \sin \frac{n\pi x}{p} \right). \quad (4.4)$$

It is a well-known theorem that if $y = f(x)$ is a periodic function with period $2p$ and $f'(x)$ is continuous on $[-p, p]$ except at finitely many points, then at each point x the Fourier series for $f(x)$ converges and

$$\frac{1}{2}a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{p} + b_n \sin \frac{n\pi x}{p} \right) = \frac{1}{2} \left(\lim_{z \rightarrow x^+} f(z) + \lim_{z \rightarrow x^-} f(z) \right).$$

In fact, if the series $\sum_{n=1}^{\infty} (|a_n| + |b_n|)$ converges, then the Fourier series converges uniformly on $(-\infty, \infty)$.

Example 4.3.3 Let $f(x) = \begin{cases} -x, & -1 \leq x < 0 \\ 1, & 0 \leq x < 1 \\ f(x-2), & x \geq 1 \end{cases}$. Compute and graph the first few partial sums of the Fourier series for $f(x)$.

Solution We begin by clearing all prior definitions of f . We then define the piecewise function $f(x)$ and graph $f(x)$ on the interval $[-1, 5]$ in Figure 4.22.

```
Clear[f]
f[x_] := 1;/; 0 ≤ x < 1
f[x_] := -x;/; -1 ≤ x < 0
f[x_] := f[x-2]/; x ≥ 1
graphf = Plot[f[x], {x, -1, 5}]
```

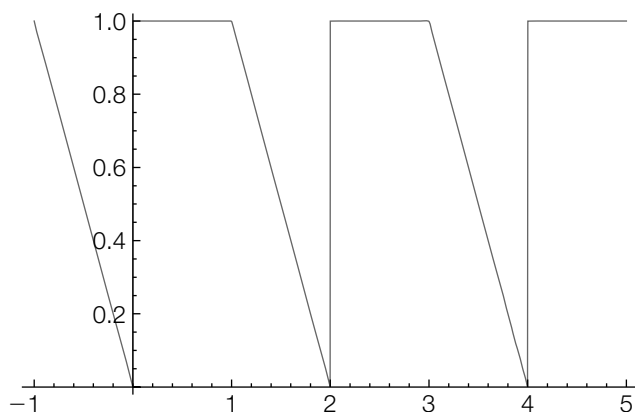


FIGURE 4.22

Plot of a few periods of $f(x)$

The Fourier series coefficients are computed with the integral formulas in equation (4.2). Executing the following commands defines p to be 1, $a[0]$ to be an approximation of the integral $a_0 = \frac{1}{p} \int_{-p}^p f(x) dx$, $a[n]$ to be an approximation of the integral $a_n = \frac{1}{p} \int_{-p}^p f(x) \cos \frac{n\pi x}{p} dx$, and $b[n]$ to be an approximation of the integral $b_n = \frac{1}{p} \int_{-p}^p f(x) \sin \frac{n\pi x}{p} dx$.

```
Clear[a, b, fs, L]
L = 1;
a[0] =  $\frac{\text{NIntegrate}[f[x], \{x, -L, L\}]}{2L}$ 
0.75
a[n_] :=  $\frac{\text{NIntegrate}[f[x] \text{Cos}[\frac{n\pi x}{L}], \{x, -L, L\}]}{L}$ 
b[n_] :=  $\frac{\text{NIntegrate}[f[x] \text{Sin}[\frac{n\pi x}{L}], \{x, -L, L\}]}{L}$ 
```

A table of the coefficients $a[i]$ and $b[i]$ for $i = 1, 2, 3, \dots, 10$ is generated with **Table** and named **coeffs**. Several error messages (which are not displayed here for length considerations) are generated because of the discontinuities, but the resulting approximations are satisfactory for our purposes. The elements in the first column of the table represent the a_i 's and those in the second column represents the b_i 's. Notice how the elements of the table are extracted using double brackets with **coeffs**.

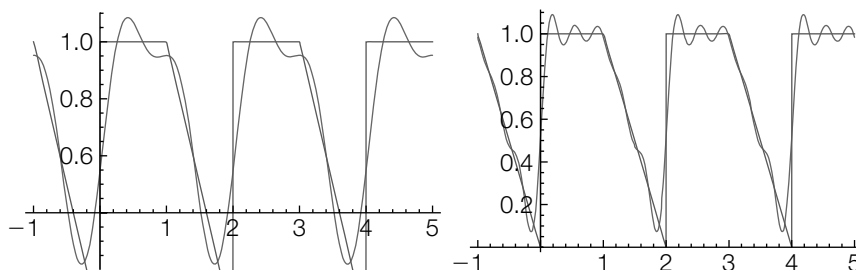
```
coeffs = Table[{a[i], b[i]}, {i, 1, 10}];
TableForm[coeffs]
-0.202642      0.31831
0.             0.159155
-0.0225158    0.106103
0.             0.0795775
-0.00810569   0.063662
0.             0.0530516
-0.00413556   0.0454728
0.             0.0397887
-0.00250176   0.0353678
-1.0668549377257364t*-16 0.031831
```

The first element of the list is extracted with **coeffs[[1]]**.

```
coeffs[[1]]
{-0.202642, 0.31831}
```

The first element of the second element of **coeffs** and the second element of the third element of **coeffs** are extracted with **coeffs[[2,1]]** and **coeffs[[3,2]]**, respectively.

```
coeffs[[2, 1]]
0.
coeffs[[3, 2]]
0.106103
```

**FIGURE 4.23**

The first few terms of a Fourier series for a periodic function plotted with the function

After the coefficients are calculated, the n th partial sum of the Fourier series is obtained with `Sum`. The k th term of the Fourier series, $a_k \cos(k\pi x) + b_k \sin(k\pi x)$, is defined in `fs`. Hence, the n th partial sum of the series is given by

$$a_0 + \sum_{k=1}^n [a_k \cos(k\pi x) + b_k \sin(k\pi x)] = a[0] + \sum_{k=1}^n fs[k, x],$$

which is defined in `fourier` using `Sum`. We illustrate the use of `fourier` by finding `fourier[2,x]` and `fourier[3,x]`.

```
fs[k_, x_] := coeffs[[k, 1]]Cos[kπx] + coeffs[[k, 2]]Sin[kπx]
fourier[n_, x_] := a[0] + Sum[fs[k, x], {k, 1, n}]
fourier[2, x]
0.75 - 0.202642Cos[πx] + 0.159155Sin[2πx]
fourier[3, x]
0.75 - 0.202642Cos[πx] + 0.159155Sin[2πx] - 0.0225158Cos[3πx] + 0.31831Sin[πx]
```

To see how the Fourier series approximates the periodic function, we plot the function simultaneously with the Fourier approximation for $n = 2$ and $n = 5$. The results are displayed together using `GraphicsArray` in Figure 4.23.

```
graphtwo = Plot[fourier[2, x], {x, -1, 5}, PlotStyle -> GrayLevel[0.4];
bothtwo = Show[graphtwo, graphf];
graphfive = Plot[fourier[5, x], {x, -1, 5}, PlotStyle -> GrayLevel[0.4];
bothfive = Show[graphfive, graphf];
Show[GraphicsRow[{bothtwo, bothfive}]]
```

Application: The One-Dimensional Heat Equation

A typical problem in applied mathematics that involves the use of Fourier series is that of the **one-dimensional heat equation**. The boundary value

problem that describes the temperature in a uniform rod with insulated surface is

$$\begin{aligned} k \frac{\partial^2 u}{\partial x^2} &= \frac{\partial u}{\partial t}, \quad 0 < x < a, \quad t > 0, \\ u(0, t) &= T_0, \quad t > 0, \\ u(a, t) &= T_a, \quad t > 0, \text{ and} \\ u(x, 0) &= f(x), \quad 0 < x < a. \end{aligned} \quad (4.5)$$

In this case, the rod has “fixed end temperatures” at $x = 0$, and $x = a$ and $f(x)$ is the initial temperature distribution. The solution to the problem is

$$u(x, t) = T_0 + \underbrace{\frac{1}{a} (T_a - T_0) x}_{v(x)} + \sum_{n=1}^{\infty} b_n \sin(\lambda_n x) e^{-\lambda_n^2 kt}, \quad (4.6)$$

where

$$\lambda_n = n\pi/a \quad \text{and} \quad b_n = \frac{2}{a} \int_0^a (f(x) - v(x)) \sin \frac{n\pi x}{a} dx,$$

and is obtained through separation of variables techniques. The coefficient b_n in the solution equation (4.6) is the Fourier series coefficient b_n of the function $f(x) - v(x)$, where $v(x)$ is the **steady-state temperature**.

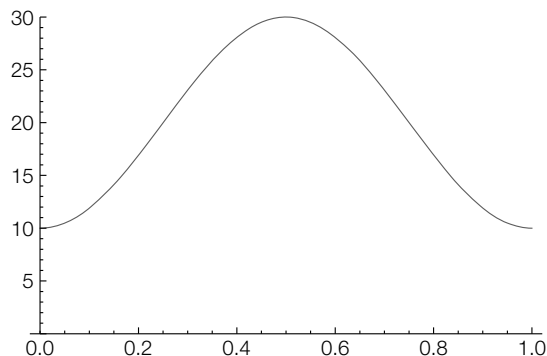
Example 4.3.4

$$\text{Solve } \begin{cases} \frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}, \quad 0 < x < 1, \quad t > 0, \\ u(0, t) = 10, \quad u(1, t) = 10, \quad t > 0, \\ u(x, 0) = 10 + 20 \sin^2 \pi x. \end{cases}$$

Solution In this case, $a = 1$ and $k = 1$. The fixed end temperatures are $T_0 = T_a = 10$, and the initial heat distribution is $f(x) = 10 + 20 \sin^2 \pi x$. The steady-state temperature is $v(x) = 10$. The function $f(x)$ is defined and plotted in Figure 4.24. Also, the steady-state temperature, $v(x)$, and the eigenvalue are defined. Finally, **Integrate** is used to define a function that will be used to calculate the coefficients of the solution.

```
Clear[f]
f[x_] := 10 + 20Sin[πx]^2
Plot[f[x]; {x, 0, 1}, PlotRange → {0, 30}]
v[x_] := 10
lambda[n_] :=  $\frac{n\pi}{4}$ 
b[n_] := b[n] =  $\int_0^4 (f[x] - v[x]) \sin\{[\frac{n\pi x}{4}]\} dx$ 
```

Notice that $b[n]$ is defined using the form $b[n_] := b[n] = \dots$ so that Mathematica “remembers” the values of $b[n]$ computed and thus avoids recomputing previously computed values. In the following table, we compute exact and approximate values of $b[1], \dots, b[10]$.

**FIGURE 4.24**Graph of $f(x) = 10 + 20 \sin^2 \pi x$ **Table[{n, b[n], b[n]/N}, {n, 1, 10}]/TableForm**

1	$\frac{5120}{63\pi}$	25.869
2	0	0.
3	$\frac{1024}{33\pi}$	9.87725
4	0	0.
5	$\frac{1024}{39\pi}$	8.35767
6	0	0.
7	$\frac{1024}{21\pi}$	15.5214
8	0	0.
9	$-\frac{5120}{153\pi}$	-10.6519
10	0	0.

Let $S_m = b_m \sin(\lambda_m x) e^{-\lambda_m^2 t}$. Then, the desired solution, $u(x, t)$, is given by

$$u(x, t) = v(x) + \sum_{m=1}^{\infty} S_m.$$

Let $u(x, t, n) = v(x) + \sum_{m=1}^n S_m$. Notice that $u(x, t, n) = u(x, t, n-1) + S_n$. Consequently, approximations of the solution to the heat equation are obtained recursively taking advantage of Mathematica's ability to compute recursively. The solution is first defined for $n = 1$ by $u[x, t, 1]$. Subsequent partial sums, $u[x, t, n]$, are obtained by adding the n th term of the series, S_n , to $u[x, t, n-1]$.

```
u[x_, t_, 1] := v[x] + b[1]Sin[lambda[1]x]Exp[-lambda[1]^2t]
u[x_, t_, n_] := u[x, t, n-1] + b[n]Sin[lambda[n]x]Exp[-lambda[n]^2t]
```

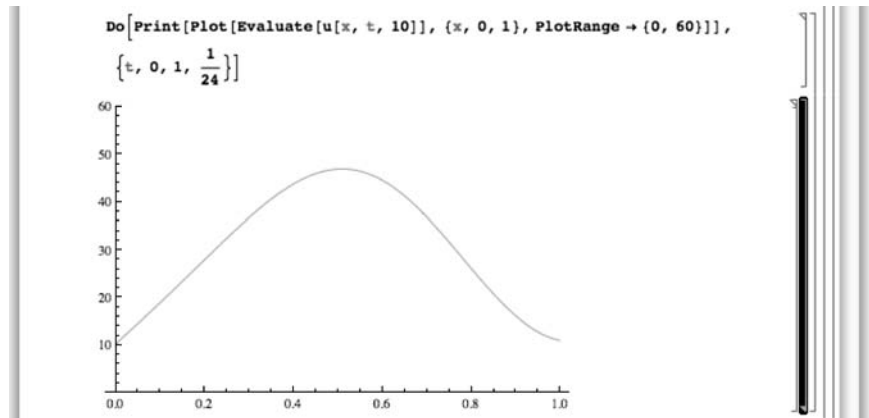
By defining the solution in this manner, a table can be created that includes the partial sums of the solution. In the following table, we compute the first, fourth, and seventh partial sums of the solution to the problem.

Table[u[x, t, n], {n, 1, 7, 3}];

To generate graphics that can be animated, we use a **Do** loop. The 10th partial sum of the solution is plotted for $t = 0$ to $t = 1$ using a step-size in t of $1/24$. Remember that $u[x,t,n]$ is determined with a **Table** command, so **Evaluate** must be used in the **Do** command so that Mathematica first computes the solution u and then evaluates u at the particular values of x . Otherwise, u is recalculated for each value of x . The plots of the solution obtained can be animated as indicated in the following screen shot.

```
Do[Print[Plot[Evaluate[u[x, t, 10]], {x, 0, 1}, PlotRange -> {0, 60}]],  
  {t, 0, 1, 1/24}]
```

Alternatively, we may generate several graphics and display the resulting set of graphics as a **GraphicsArray**. We plot the 10th partial sum of the solution for $t = 0$ to $t = 1$ using a step-size of $1/15$. The resulting 16 graphs are named **graphs**, which are then partitioned into four element subsets with **Partition** and named **toshow**. We then use **Show** and **GraphicsGrid** to display **toshow** in Figure 4.25.



```
graphs = Table[Plot[Evaluate[u[x, t, 10]], {x, 0, 1}, Ticks -> None,  
  PlotRange -> {0, 60}, DisplayFunction -> Identity], {t, 0, 1, 1/15}];  
toshow = Partition[graphs, 4];  
Show[GraphicsGrid[toshow]]
```

Fourier series and generalized Fourier series arise in too many applications to list. Examples using them illustrate Mathematica's power to manipulate lists, symbolics, and graphics.

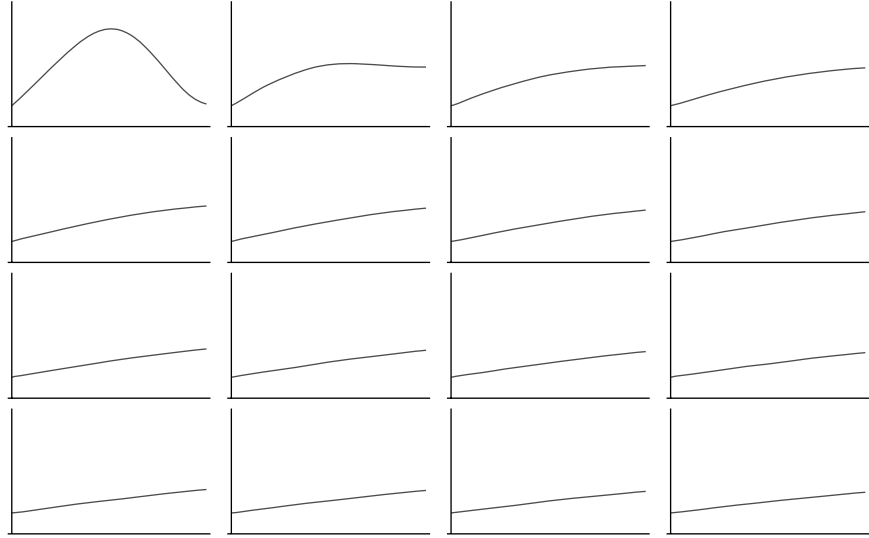


FIGURE 4.25

Temperature distribution in a uniform rod with insulated surface

For a classic approach to the subject, see Graff's *Wave Motion in Elastic Solids*, [10].

Application: The Wave Equation on a Circular Plate

The vibrations of a circular plate satisfy the equation

$$D \nabla^4 w(r, \theta, t) + \rho b \frac{\partial^2 w(r, \theta, t)}{\partial t^2} = q(r, \theta, t), \quad (4.7)$$

where $\nabla^4 w = \nabla^2 \nabla^2 w$ and ∇^2 is the **Laplacian in polar coordinates**, which is defined by

$$\nabla^2 = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}.$$

Assuming no forcing so that $q(r, \theta, t) = 0$ and $w(r, \theta, t) = W(r, \theta)e^{-i\omega t}$, equation (4.7) can be written as

$$\nabla^4 W(r, \theta) - \beta^4 W(r, \theta) = 0, \quad \beta^4 = \omega^2 \rho b / D. \quad (4.8)$$

For a clamped plate, the boundary conditions are $W(a, \theta) = \partial W(a, \theta) / \partial r = 0$, and after *much work* (see [10]) the **normal modes** are found to be

$$W_{nm}(r, \theta) = \left[J_n(\beta_{nm}r) - \frac{J_n(\beta_{nm}a)}{I_n(\beta_{nm}a)} I_n(\beta_{nm}r) \right] \begin{pmatrix} \sin n\theta \\ \cos n\theta \end{pmatrix}. \quad (4.9)$$

In equation (4.9), $\beta_{nm} = \lambda_{nm}/a$, where λ_{nm} is the m th solution of

$$I_n(x)J_n'(x) - J_n(x)I_n'(x) = 0, \quad (4.10)$$

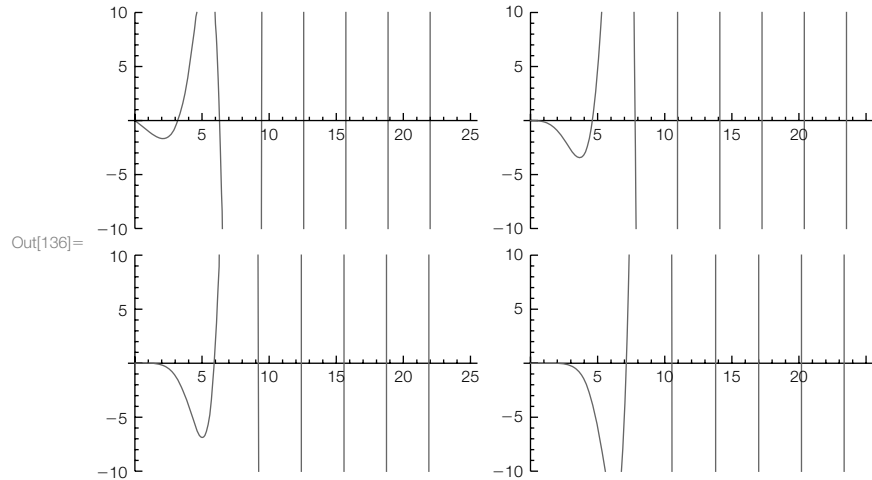


FIGURE 4.26

Plot of $I_n(x)J_n'(x) - J_n(x)I_n'(x)$ for $n = 0$ and 1 in the first row and $n = 2$ and 3 in the second row

See Example 4.2.4.

where $J_n(x)$ is the Bessel function of the first kind of order n and $I_n(x)$ is the **modified Bessel function of the first kind** of order n , related to $J_n(x)$ by $i^n I_n(x) = J_n(ix)$.

The Mathematica command `BesselI[n,x]` returns $I_n(x)$.

Example 4.3.5 Graph the first few normal modes of the clamped circular plate.

Solution We must determine the value of λ_{nm} for several values of n and m , so we begin by defining `eqn[n][x]` to be $I_n(x)J_n'(x) - J_n(x)I_n'(x)$. The m th solution of equation (4.10) corresponds to the m th zero of the graph of `eqn[n][x]`, so we graph `eqn[n][x]` for $n = 0, 1, 2$, and 3 with `Plot` in Figure 4.26.

```
eqn[n_][x_]:=BesselI[n,x]D[BesselJ[n,x],x]-BesselJ[n,x]D[BesselI[n,x],x]
```

The result of the `Table` and `Plot` command is a list of length four, which is verified with `Length[p1]`.

```
p1 = Table[Plot[Evaluate[eqn[n][x]], {x, 0, 25}, PlotRange -> {-10, 10}], {n, 0, 3}];
```

so we use `Partition` to create a 2×2 array of graphics that is displayed using `Show` and `GraphicsGrid`.

```
p2 = Show[GraphicsGrid[Partition[p1, 2]]]
```

To determine λ_{nm} , we use `FindRoot`. Recall that to use `FindRoot` to solve an equation, an initial approximation of the solution must be given. For example,

```
l01 = FindRoot[eqn[0][x] == 0, {x, 3.04}]
{x → 3.19622}
```

approximates λ_{01} , the first solution of equation (4.10) if $n = 0$. However, the result of `FindRoot` is a list. The specific value of the solution is the second part of the first part of the list, `lambda01`, extracted from the list with `Part ([[...]])`.

```
l01[[1, 2]]
3.19622
```

Thus,

We use the graphs in Figure 4.26 to obtain initial approximations of each solution.

```
l0s = Map[FindRoot[eqn[0][x] == 0, {x, #}][[1, 2]] &,
{3.04, 6.2, 9.36, 12.5, 15.7}]
{3.19622, 6.30644, 9.4395, 12.5771, 15.7164}
```

approximates the first five solutions of equation (4.10) if $n = 0$ and then returns the specific value of each solution. We use the same steps to approximate the first five solutions of equation (4.10) if $n = 1, 2$, and 3 .

```
l1s = Map[FindRoot[eqn[1][x] == 0, {x, #}][[1, 2]] &,
{4.59, 7.75, 10.9, 14.1, 17.2}]
{4.6109, 7.79927, 10.9581, 14.1086, 17.2557}
l2s = Map[FindRoot[eqn[2][x] == 0, {x, #}][[1, 2]] &,
{5.78, 9.19, 12.4, 15.5, 18.7}]
{5.90568, 9.19688, 12.4022, 15.5795, 18.744}
l3s = Map[FindRoot[eqn[3][x] == 0, {x, #}][[1, 2]] &,
{7.14, 10.5, 13.8, 17, 20.2}]
{7.14353, 10.5367, 13.7951, 17.0053, 20.1923}
```

All four lists are combined together in `ls`.

```
ls = {l0s, l1s, l2s, l3s};
Short[ls]
{{3.19622, 6.30644, <<18>>, 12.5771, 15.7164}, <<2>>, {{<<1>>}}}
```

For $n = 0, 1, 2$, and 3 and $m = 1, 2, 3, 4$, and 5 , λ_{nm} is the m th part of the $(n + 1)$ st part of `ls`.

Observe that the value of a does not affect the shape of the graphs of the normal modes, so we use $a = 1$ and then define β_{nm} .

```
a = 1;
β[n_, m_] := ls[[n + 1, m]]/a
```

ws is defined to be the sine part of equation (4.9)

```
ws[n_, m_][r_, θ_] :=  
(BesselJ[n, β[n, m]r] - BesselJ[n, β[n, m]a]/BesselI[n, β[n, m]a]  
  BesselI[n, β[n, m]r])Sin[nθ]
```

and wc to be the cosine part.

```
wc[n_, m_][r_, θ_] :=  
(BesselJ[n, β[n, m]r] - BesselJ[n, β[n, m]a]/BesselI[n, β[n, m]a]  
  BesselI[n, β[n, m]r])Cos[nθ]
```

We use ParametricPlot3D to plot ws and wc. For example,

```
ParametricPlot3D[{rCos[θ], rSin[θ], ws[3, 4][r, θ]}, {r, 0, 1}, {θ, -Pi, Pi},  
  PlotPoints → 60]
```

graphs the sine part of $W_{34}(r, \theta)$ shown in Figure 4.27. We use Table together with ParametricPlot3D followed by Show and GraphicsGrid to graph the sine part of $W_{nm}(r, \theta)$ for $n = 0, 1, 2$, and 3 and $m = 1, 2, 3$, and 4 shown in Figure 4.28.

```
ms = Table[ParametricPlot3D[{rCos[θ], rSin[θ], ws[n, m][r, θ]},  
  {r, 0, 1}, {θ, -Pi, Pi},  
  DisplayFunction → Identity, PlotPoints → 30, BoxRatios → {1, 1, 1}],  
  {n, 0, 3}, {m, 1, 4}];  
Show[GraphicsGrid[ms]]
```

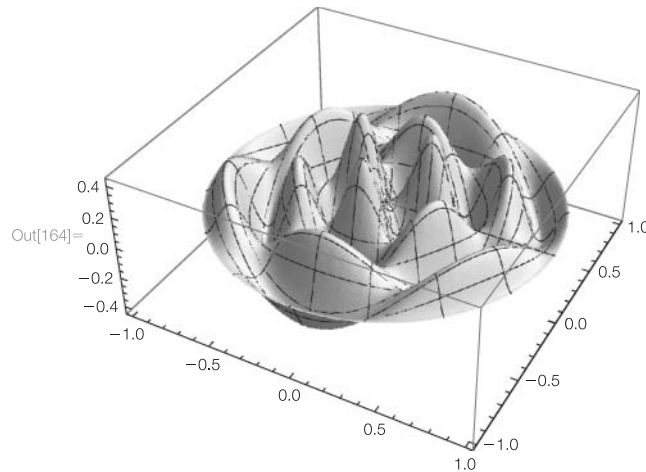
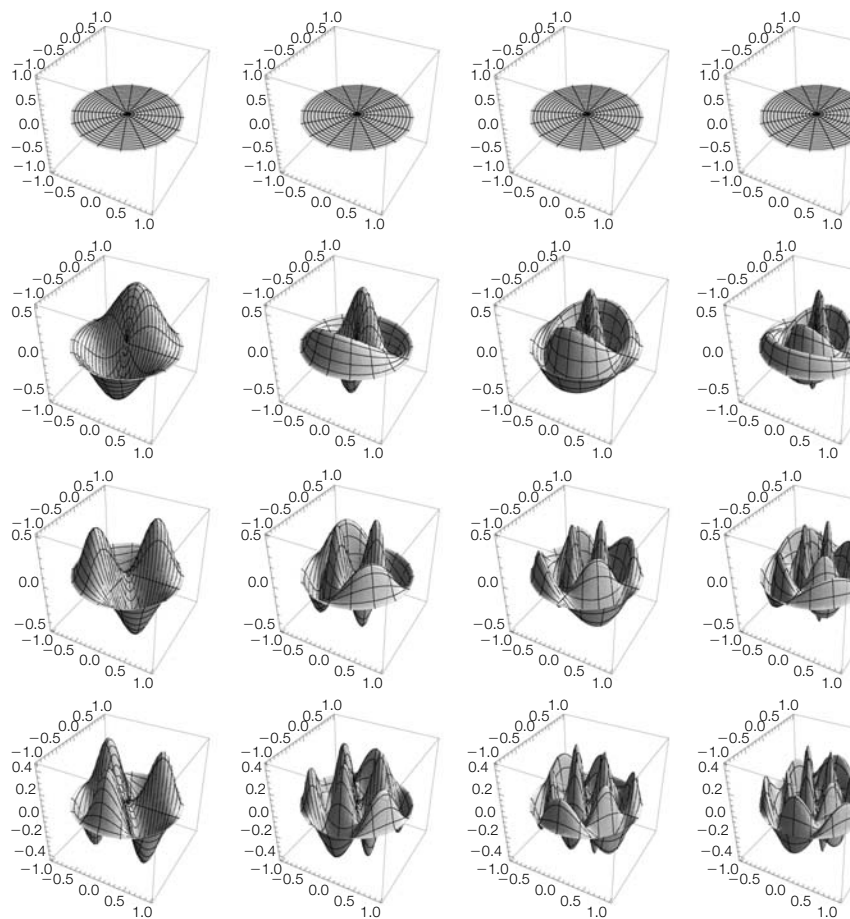


FIGURE 4.27

The sine part of $W_{34}(r, \theta)$

**FIGURE 4.28**

The sine part of $W_{nm}(r, \theta)$: $n = 0$ in row 1, $n = 1$ in row 2, $n = 2$ in row 3, and $n = 3$ in row 4 ($m = 1$ to 4 from left to right in each row)

Identical steps are followed to graph the cosine part shown in Figure 4.29.

```
mc = Table[ParametricPlot3D[{rCos[θ], rSin[θ], wc[n, m][r, θ]},  
  {r, 0, 1}, {θ, -Pi, Pi},  
  DisplayFunction → Identity, PlotPoints → 30, BoxRatios → {1, 1, 1},  
  {n, 0, 3}, {m, 1, 4}];  
Show[GraphicsGrid[mc]]
```

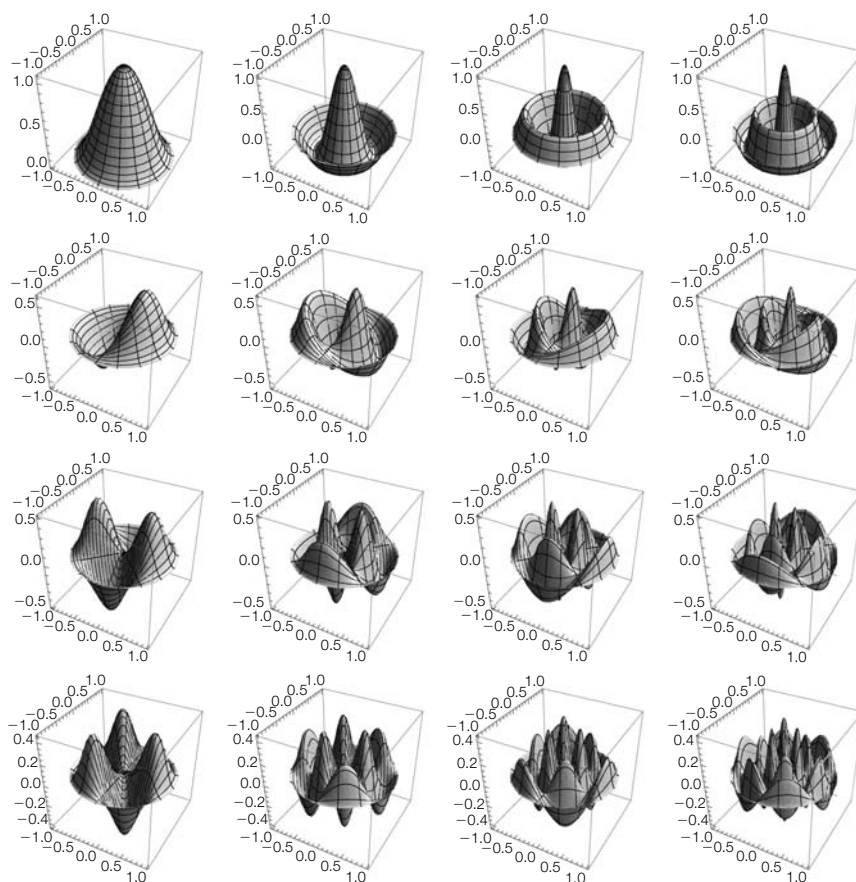


FIGURE 4.29

The cosine part of $W_{nm}(r, \theta)$: $n = 0$ in row 1, $n = 1$ in row 2, $n = 2$ in row 3, and $n = 3$ in row 4 ($m = 1$ to 4 from left to right in each row)

See references such as Barnsley's *Fractals Everywhere* [4] or Devaney and Keen's *Chaos and Fractals* [6] for detailed discussions regarding many of the topics briefly described in this section.

$f_c(x) = x^2 + c$ is the special case of $p = 2$ for $f_{p,c}(x) = x^p + c$.

4.3.3 The Mandelbrot Set and Julia Sets

In Examples 4.1.9, 4.2.5, and 4.2.7 we illustrated several techniques for plotting bifurcation diagrams and Julia sets.

Let $f_c(x) = x^2 + c$. In Example 4.2.5, we generated the c -values when plotting the bifurcation diagram of f_c . Depending on how you think, some approaches may be easier to understand than others. With the exception of very serious calculations, the differences in the time needed to carry out the computations may be minimal, so we encourage you to follow the approach that you understand. Learn new techniques as needed.

Example 4.3.6 (Dynamical Systems). For example, entering

Compare the approach here with the approach used in Example 4.2.5.

```
Clear[f, h]
f[c_][x_] := x^2 + c/N;
```

defines $f_c(x) = x^2 + c$, so

```
Nest[f[-1], x, 3]
-1. + (-1. + (-1. + x^2)^2)
```

computes $f_{-1}^3(x) = (f_{-1} \circ f_{-1} \circ f_{-1})(x)$ and

```
Table[Nest[f[1/4], 0, n], {n, 101, 200}]/Short
{0.490693, 0.490779, <96>, 0.495148, 0.495171}
```

returns a list of $f_{1/4}^n(0)$ for $n = 101, 102, \dots, 200$. Thus,

```
lgtable = Table[{c, Nest[f[c], 0, n]},
{c, -2, 1/4, 9/(4 * 299)}, {n, 101, 200}];
Length[lgtable]
300
```

returns a list of lists of $f_c^n(0)$ for $n = 101, 102, \dots, 200$ for 300 equally spaced values of c between -2 and 1 . The list `lgtable` is converted to a list of points with `Flatten` and plotted with `ListPlot`. See Figure 4.30 and compare this result to the result obtained in Example 4.2.5.

```
toplot = Flatten[lgtable, 1];
ListPlot[toplot]
```

For a given complex number c , the **Julia set**, J_c , of $f_c(x) = x^2 + c$ is the set of complex numbers, $z = a + bi$, a, b real, for which the sequence $z, f_c(z) = z^2 + c$,

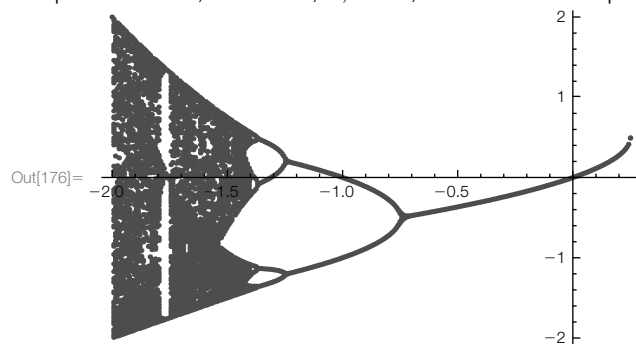


FIGURE 4.30

Another bifurcation diagram for f_c

We use the notation $f^n(x)$ to represent the composition $(\underbrace{f \circ f \circ \dots \circ f}_n)(x)$.

$f_c(f_c(z)) = (z^2 + c)^2 + c, \dots, f_c^n(z), \dots$, does *not* tend to ∞ as $n \rightarrow \infty$:

$$J_c = \left\{ z \in \mathbb{C} \mid z, z^2 + c, (z^2 + c)^2 + c, \dots \not\rightarrow \infty \right\}.$$

Using a dynamical system, setting $z = z_0$ and computing $z_{n+1} = f_c(z_n)$ for large n can help us determine if z is an element of J_c . In terms of a composition, computing $f_c^n(z)$ for large n can help us determine if z is an element of J_c .

Example 4.3.7 (Julia Sets). Plot the Julia set of $f_c(x) = x^2 + c$ if $c = -0.122561 + 0.744862i$.

Solution

As before, all error messages have been deleted.

You do not need to redefine $f_c(x)$ if you have already defined it during your current Mathematica session.

After defining $f_c(x) = x^2 + c$, we use `Table` together with `Nest` to compute ordered triples of the form $(x, y, f_{-0.122561+0.744862i}^{200}(x + iy))$ for 150 equally spaced values of x between $-3/2$ and $3/2$ and 150 equally spaced values of y between $-3/2$ and $3/2$.

```
Clear[f, h]
f[c_][x_] := x^2 + c/N;
g1 = Table[{x, y, Nest[f[-0.12256117 + .74486177I], x + Iy, 200]},
{x, -3/2, 3/2, 3/149}, {y, -3/2, 3/2, 3/149}];
g2 = Flatten[g1, 1];
```

We remove those elements of `g2` for which the third coordinate is `Overflow[ ]` with `Select`,

```
g3 = Select[g2, Not[#[[3]] === Overflow[ ]]&];
```

extract a list of the first two coordinates, (x, y) , from the elements of `g3`,

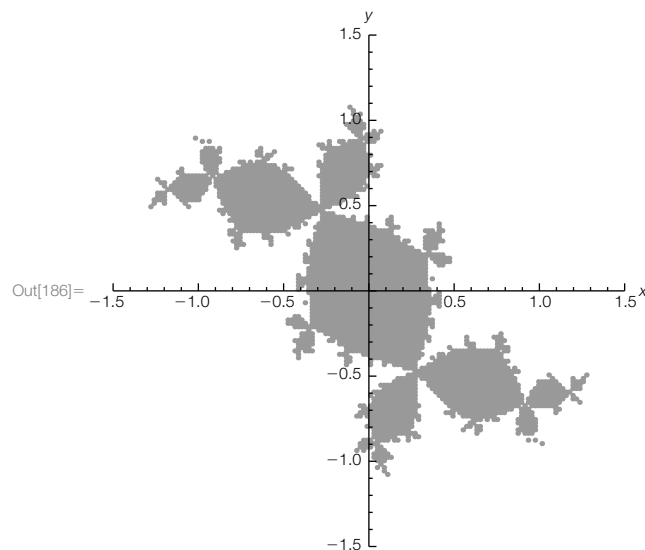
```
g4 = Map[{#[[1]], #[[2]]}&, g3];
```

and plot the resulting list of points in Figure 4.31 using `ListPlot`.

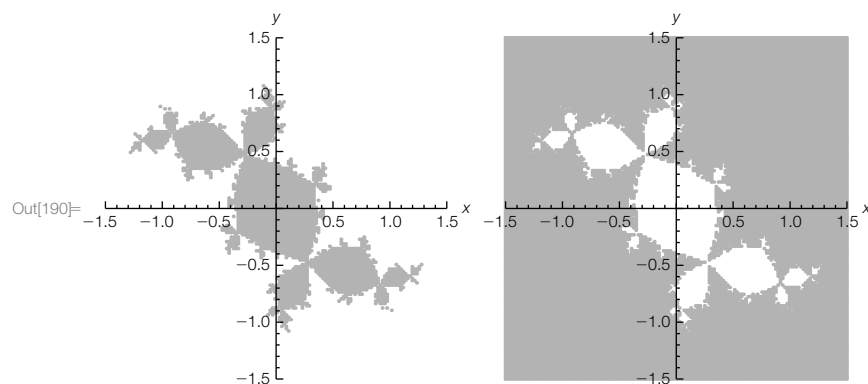
```
lp1 = ListPlot[g4, PlotRange -> {{-3/2, 3/2}, {-3/2, 3/2}},
AxesLabel -> {"x", "y"}, AspectRatio -> Automatic]
```

We can invert the image as well with the following commands. In the end result, we show the Julia set and its inverted image in Figure 4.32

```
g3b = Select[g2, #[[3]] === Overflow[ ]&];
g4b = Map[{#[[1]], #[[2]]}&, g3b];
lp2 = ListPlot[g4b, PlotRange -> {{-3/2, 3/2}, {-3/2, 3/2}},
AxesLabel -> {"x", "y"}, AspectRatio -> Automatic,
DisplayFunction -> Identity];
j1 = Show[GraphicsRow[{lp1, lp2}]]
```

**FIGURE 4.31**

Filled Julia set for f_c

**FIGURE 4.32**

Filled Julia set for f_c on the left; the inverted set on the right

Of course, one can consider functions other than $f_c(x) = x^2 + c$ as well as rearrange the order in which we carry out the computations. You have even greater control over your graphics if you use graphics primitives such as `Point`.

Example 4.3.8 (Julia Sets). Plot the Julia set for $f_c(z) = z^2 - cz$ if $c = 0.737369 + 0.67549i$.

Solution We initially proceed as in Example 4.3.7.

As before, all error messages have been deleted.

```
Clear[f, h]
f[c_][x_] := x ^ 2 - cx//N;
g1 = Table[{x, y, Nest[f[0.737369 + 0.67549i], x + Iy, 200]},
{x, -3/2, 3/2, 3/149}, {y, -3/2, 3/2, 3/149}];
g2 = Flatten[g1, 1];
g3 = Select[g2, Not[#[[3]] === Overflow[]]&];
```

After removing the points that result in an `Overflow[]` error message, we code the remaining ones according to their distance from the origin.

```
h[{x_, y_, z_}] := {x, y, Min[Abs[z], 0.5]}
g4 = Map[h, g3];

g5 = Table[{PointSize[0.005], GrayLevel[g4[[i, 3]]/0.5],
Point[{g4[[i, 1]], g4[[i, 2]]}], {i, 1, Length[g4]}};
```

The results are shown in Figure 4.33.

```
lp1 = ListPlot[g4, PlotRange -> {{-3/2, 3/2}, {-3/2, 3/2}},
AxesLabel -> {"x", "y"}, AspectRatio -> Automatic]
Show[Graphics[g5], PlotRange -> >{{-1.2, 1.75}, {-0.7, 1.4}},
AspectRatio -> Automatic]
```

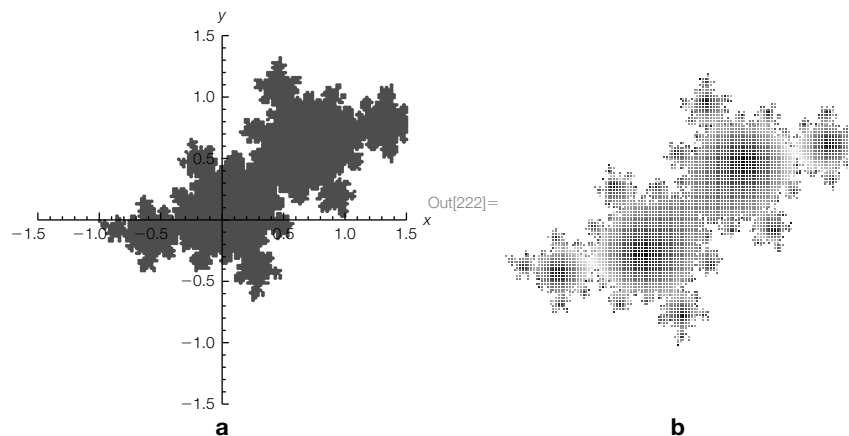


FIGURE 4.33

(a) The Julia set. (b) The lightest points (a, b) are the ones for which $\left|f_{0.737369+0.67549i}^{200}(z)\right|$ is the largest

Example 4.3.9 (The Ikeda Map). The **Ikeda map** is defined by

$$F(x, y) = \langle \gamma + \beta (x \cos \tau - y \sin \tau), \beta (x \sin \tau + y \cos \tau) \rangle, \quad (4.11)$$

where $\tau = \mu - \alpha / (1 + x^2 + y^2)$. If $\beta = 0.9$, $\mu = 0.4$, and $\alpha = 4.0$, plot the *basins of attraction* for F if $\gamma = 0.92$ and $\gamma = 1.0$.

Solution The *basins of attraction* for F are the set of points (x, y) for which $\|F^n(x, y)\| \rightarrow \infty$ as $n \rightarrow \infty$.

After defining $f[\gamma][x, y]$ to be equation (4.11) and then $\beta = 0.9$, $\mu = 0.4$, and $\alpha = 4.0$, we use **Table** followed by **Flatten** to define **pts** to be the list of 40,000 ordered pairs (x, y) for 200 equally spaced values of x between -2.3 and 1.3 and 200 equally spaced values of y between -2.8 and 0.8 .

```
f[γ-][[x-, y-]] := {γ + β(xCos[μ - α/(1 + x^2 + y^2)] - y
Sin[μ - α/(1 + x^2 + y^2)]), β(xSin[μ - α/(1 + x^2 + y^2)] + y
Cos[μ - α/(1 + x^2 + y^2)])}
β = 0.9; μ = 0.4; α = 4.0;
pts = Flatten[Table[{x, y}, {x, -2.3, 1.3, 3.6/199}, {y, -2.8, 0.8, 3.6/199}], 1];
```

In **I1**, we use **Map** to compute $(x, y, F_{.92}^{200}(x, y))$ for each (x, y) in **pts**. In **pts2**, we use the graphics primitive **Point** and shade the points according to the maximum value of $\|F^{200}(x, y)\|$ —those (x, y) for which $F^{200}(x, y)$ is closest to the origin are darkest; the point (x, y) is shaded lighter as the distance of $F^{200}(x, y)$ from the origin increases. (See Figure 4.34(a).)

```
I1 = Map[{#[[1]], #[[2]], Nest[f[.92], {#[[1]], #[[2]]}, 200]}&, pts];
g[{x-, y-, z-}] := {x, y, Sqrt[z[[1]]^2 + z[[2]]^2]};
I2 = Map[g, I1];
maxI2 = Table[I2[[i, 3]], {i, 1, Length[I2]}/Max
4.33321
pts2 = Table[{GrayLevel[I2[[i, 3]]/(maxI2)], Point[{I2[[i, 1]], I2[[i, 2]]}],
{i, 1, Length[I2]}}];
ik1 = Show[Graphics[pts2], AspectRatio -> 1];
```

For $\gamma = 1.0$, we proceed in the same way. The final results are shown in Figure 4.34(b).

```
I1 = Map[{#[[1]], #[[2]], Nest[f[1.0], {#[[1]], #[[2]]}, 200]}&, pts];
I2 = Map[g, I1];
maxI2 = Table[I2[[i, 3]], {i, 1, Length[I2]}/Max
4.48421
pts2 = Table[{GrayLevel[I2[[i, 3]]/maxI2], Point[{I2[[i, 1]], I2[[i, 2]]}],
{i, 1, Length[I2]}}];
ik2 = Show[Graphics[pts2], AspectRatio -> 1]
Show[GraphicsRow[{ik1, ik2}]]
```

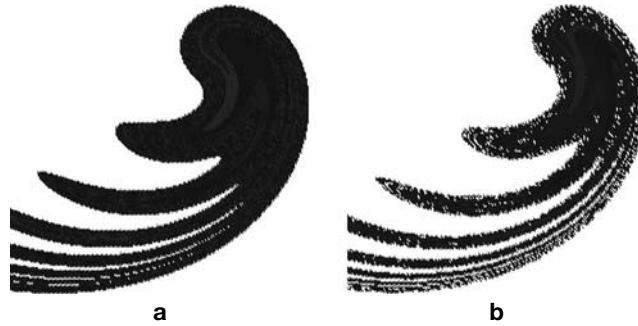


FIGURE 4.34

Basins of attraction for F if (a) $\gamma = 0.92$ and (b) $\gamma = 1.0$

The **Mandelbrot set**, M , is the set of complex numbers, $z = a + bi$, a, b real, for which the sequence $z, f_z(z) = z^2 + z, f_z(f_z(z)) = (z^2 + z)^2 + z, \dots, f_z^n(z), \dots$, does *not* tend to ∞ as $n \rightarrow \infty$:

$$M = \left\{ z \in \mathbb{C} \mid z, z^2 + z, (z^2 + z)^2 + z, \dots \not\rightarrow \infty \right\}.$$

Using a dynamical system, setting $z = z_0$ and computing $z_{n+1} = f_{z_0}(z_n)$ for large n can help us determine if z is an element of M . In terms of a composition, computing $f_z^n(z)$ for large n can help us determine if z is an element of M .

Example 4.3.10 (Mandelbrot Set). Plot the Mandelbrot set.

Solution We proceed as in Example 4.3.7 except that instead of iterating $f_c(z)$ for fixed c , we iterate $f_z(z)$.

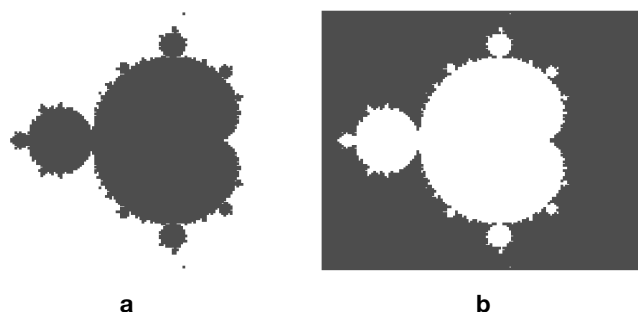
As before, all error messages have been deleted.

As with the previous examples, all Overflow[...] messages have been deleted.

```
Clear[f, h]
f[c_][x_] := x^2 + c/N;
g1 = Table[{x, y, Nest[f[x + Iy], x + Iy, 200]},
  {x, -3/2, 1, 5/(2 * 149)}, {y, -1, 1, 2/149}];
g2 = Flatten[g1, 1];
Take[g2, 5]
g3 = Select[g2, Not[#[[3]] === Overflow[ ]]&];
g4 = Map[{#[[1]], #[[2]]}&, g3];
```

The following gives us the image in Figure 4.35(a).

```
lp1 = ListPlot[g4, PlotRange -> {{-3/2, 1}, {-1, 1}}, Axes -> None,
  AspectRatio -> Automatic, PlotStyle -> PointSize[.005]];
```

**FIGURE 4.35**

Two different views of the Mandelbrot set: in (a), the black points (a, b) are the points for which $f_{a+bi}^{200}(a + bi)$ is finite; in (b), the black points (a, b) are the ones for which $f_{a+bi}^{200}(a + bi)$ is not finite

To invert the image, we use the following to obtain the result in Figure 4.35(b).

```
g3b = Select[g2, #[[3]]===Overflow[ ]&];
g4b = Map[{#[[1]], #[[2]]}&, g3b];
lp2 = ListPlot[g4b, PlotRange → {{-3/2, 1}, {-1, 1}},
  Axes → None, AspectRatio → Automatic, PlotStyle → PointSize[.005];
m1 = Show[GraphicsRow[{lp1, lp2}]]
```

In Example 4.3.10, the Mandelbrot set is obtained (or, more precisely, approximated) by repeatedly composing $f_z(z)$ for a grid of z -values and then deleting those for which the values exceed machine precision. Those values greater than $\$MaxNumber$ result in an `Overflow[ ]` message; computations with `Overflow[ ]` result in an `Indeterminate` message.

We can generalize by considering exponents other than 2 by letting $f_{p,c} = x^p + c$. The **generalized Mandelbrot set**, M_p , is the set of complex numbers, $z = a + bi$, a, b real, for which the sequence $z, f_{p,z}(z) = z^p + z, f_{p,z}(f_{p,z}(z)) = (z^p + z)^p + z, \dots, f_{p,z}^n(z), \dots$, does *not* tend to ∞ as $n \rightarrow \infty$:

$$M_p = \left\{ z \in \mathbb{C} \mid z, z^p + z, (z^p + z)^p + z, \dots \not\rightarrow \infty \right\}.$$

Using a dynamical system, setting $z = z_0$ and computing $z_{n+1} = f_p(z_n)$ for large n can help us determine if z is an element of M_p . In terms of a composition, computing $f_p^n(z)$ for large n can help us determine if z is an element of M_p .

Example 4.3.11

As with the previous examples, all error messages have been omitted.

(Generalized Mandelbrot Set). After defining $f_{p,c} = x^p + c$, we use `Table`, `Abs`, and `Nest` to compute a list of ordered triples of the form $(x, y, |f_{p,x+iy}^{100}(x+iy)|)$ for p -values from 1.625 to 2.625 spaced by equal values of 1/8 and 200 values of $x(y)$ values equally spaced between -2 and 2 , resulting in 40,000 sample points of the form $x + iy$.

```
Clear[f, p]
f[p_, c_][x_]:=x ^ p + c//N;
g1 =
Map[Table[{x, y, Abs[Nest[f[2, x + Iy], x + Iy, #]]}/N,
  {x, -1.5, 1., 5/(2 * 199)}, {y, -1., 1., 2/199}]&, {5, 10, 15, 25, 50, 100}];
g2 = Map[Flatten[#, 1]&, g1];
```

Next, we extract those points for which the third coordinate is `Indeterminate` with `Select`; ordered pairs of the first two coordinates are obtained in `g4`. The resulting list of points is plotted with `ListPlot` in Figure 4.36.

```
g3 = Table[Select[g2[[i]], Not[#[[3]]===Overflow[]]&], {i, 1, Length[g2]}];
h[{x_, y_, z_}]:={x, y};
g4 = Map[h, g3, {2}];
t1 = Table[ListPlot[g4[[i]], PlotRange -> {{-3/2, 1}, {-1, 1}},
  AspectRatio -> Automatic, DisplayFunction -> Identity], {i, 1, 6}];
Show[GraphicsGrid[Partition[t1, 3]]]
```

More detail is observed if you use the graphics primitive `Point` as shown in Figure 4.37. In this case, those points (x, y) for which $|f_{p,x+iy}^{100}(x+iy)|$ is small are shaded according to a darker `GrayLevel` than those points for which $|f_{p,x+iy}^{100}(x+iy)|$ is large.

```
h2[{x_, y_, z_}]:={GrayLevel[Min[{z, 1}]], Point[{x, y}];
g5 = Map[h2, g3, {2}];
t1 = Table[Show[Graphics[g5[[i]]], PlotRange -> {{-3/2, 1}, {-1, 1}},
  AspectRatio -> Automatic, DisplayFunction -> Identity], {i, 1, 6}];
Show[GraphicsGrid[Partition[t1, 3]]]
```

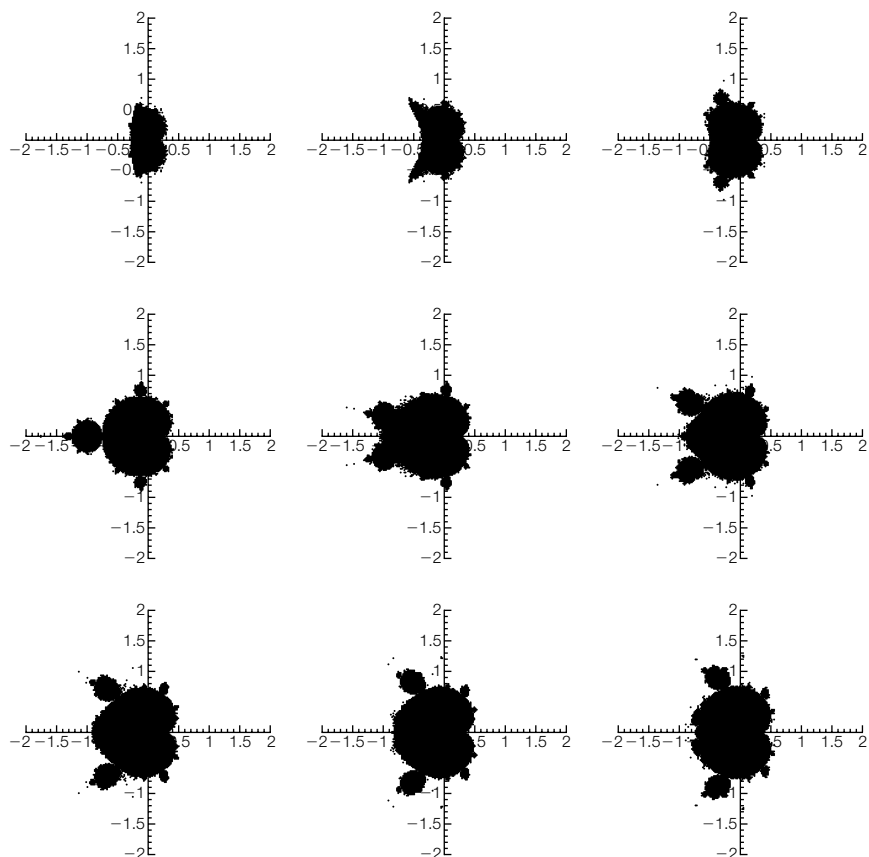
Throughout these examples, we have typically computed the iteration $f^n(z)$ for “large” n , such as values of n between 100 and 200. To indicate why we have selected those values of n , we revisit the Mandelbrot set plotted in Example 4.3.10.

Example 4.3.12

As before, all error messages have been deleted.

(Mandelbrot Set). We proceed in essentially the same way as in the previous examples. After defining $f_{p,c} = x^p + c$,

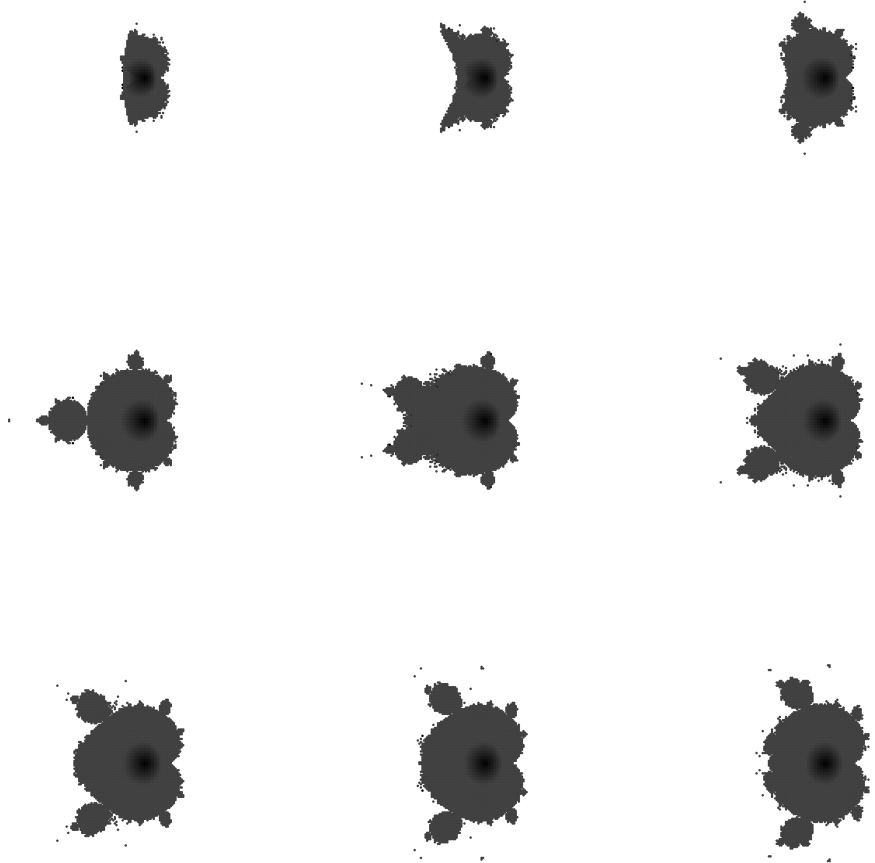
```
Clear[f, p]
f[p_, c_][x_]:=x ^ p + c//N;
```

**FIGURE 4.36**

The generalized Mandelbrot set for nine equally spaced values of p between 1.625 and 2.625

we use `Table` followed by `Map` to create a nested list. For each $n = 5, 10, 15, 25, 50$, and 100 , a nested list is formed for 200 equally spaced values of y between -1 and 1 and then 200 equally spaced values of x between -1.5 and 1 . At the bottom level of each nested list, the elements are of the form $(x, y, |f_{2,x+iy}^n(x+iy)|)$.

```
g1 =
Map[Table[{x, y, Abs[Nest[f[2, x + Iy], x + Iy, #]]}/N,
{x, -1.5, 1., 5/(2 * 199)}, {y, -1., 1., 2/199}]&,
{5, 10, 15, 25, 50, 100}];
```

**FIGURE 4.37**

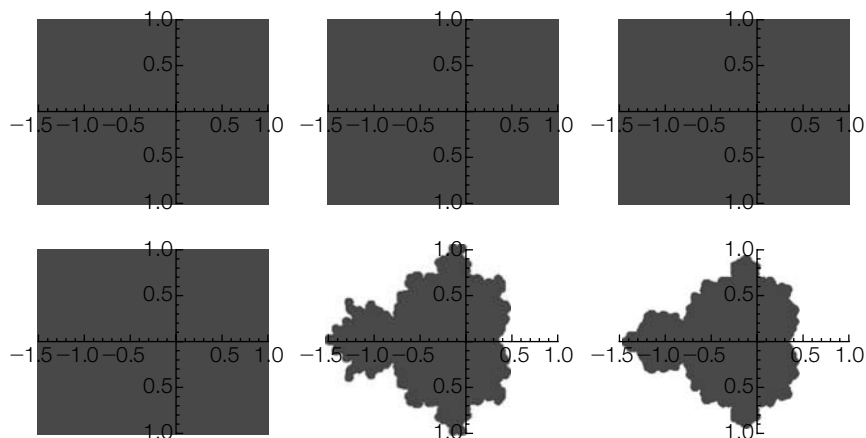
The generalized Mandelbrot set for nine equally spaced values of p between 1.625 and 2.625—the points (x, y) for which $\left|f_{p, x+iy}^{100}(x+iy)\right|$ is large are shaded lighter than those for which $\left|f_{p, x+iy}^{100}(x+iy)\right|$ is small

For each value of n , the corresponding list of ordered triples $(x, y, |f_{2, x+iy}^n(x+iy)|)$ is obtained using `Flatten`.

```
g2 = Map[Flatten[#, 1]&, g1];
```

We then remove those points for which the third coordinate, $|f_{2, x+iy}^n(x+iy)|$, is `Overflow[]` (corresponding to ∞),

```
g3 = Table[Select[g2[[i]], Not[#[[3]]===Overflow[]]&], {i, 1, Length[g2]}];
```

**FIGURE 4.38**

Without shading the points, the effects of iteration are difficult to see until the number of iterations is “large”

extract (x, y) from the remaining ordered triples,

```
h[{x_, y_, z_}] := {x, y};
g4 = Map[h, g3, {2}];
```

Fundamentally, we generated the previous plots by exceeding Mathematica’s numerical precision.

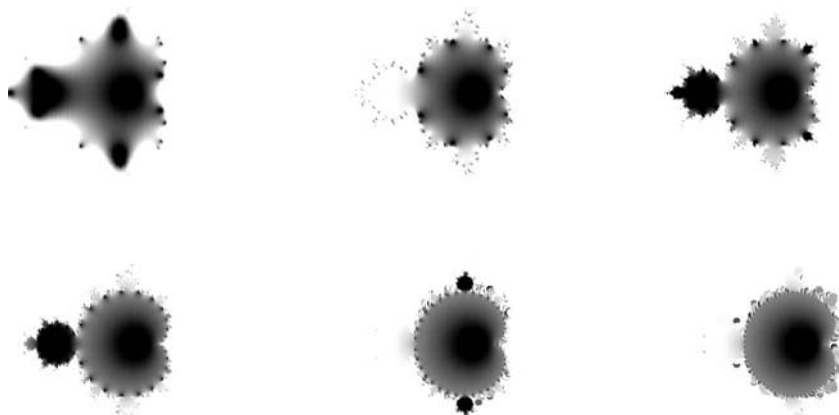
and graph the resulting sets of points using `ListPlot` in Figure 4.38. As shown in Figure 4.38, we see that Mathematica’s numerical precision (and consequently decent plots) is obtained when $n = 50$ or $n = 100$.

```
t1 = Table[ListPlot[g4[[i]], PlotRange → {{-3/2, 1}, {-1, 1}},
AspectRatio → Automatic, DisplayFunction → Identity], {i, 1, 6}];
Show[GraphicsGrid[Partition[t1, 3]]]
```

If instead, we use graphics primitives such as `Point` and then shade each point (x, y) according to $|f_{2,x+iy}^n(x+iy)|$, detail emerges quickly, as shown in Figure 4.39.

```
h2[{x_, y_, z_}] := {GrayLevel[Min[{z, 1}], Point[{x, y}]}];
g5 = Map[h2, g3, {2}];
t1 = Table[Show[Graphics[g5[[i]]], PlotRange → {{-3/2, 1}, {-1, 1}},
AspectRatio → Automatic, DisplayFunction → Identity], {i, 1, 6}];
Show[GraphicsGrid[Partition[t1, 3]]]
```

Thus, Figures 4.38 and 4.39 indicate that for examples such as these illustrated here, similar results could have been accomplished using far smaller values of n than $n = 100$ or $n = 200$. With fast machines, the differences in the time needed to perform the calculations is minimal; $n = 100$ and $n = 200$ appear to be a “safe” large value of n for well-studied examples such as these.

**FIGURE 4.39**

Using graphics primitives and shading, we see that we can use a relatively small number of iterations to visualize the Mandelbrot set

4.4 EXERCISES

1. Use Mathematica help to determine the functionality of Chop.
2. Define zeros to be the list of numbers 2.4048, 5.5201, 8.6537, 11.792, 14.931, 18.071, 21.212, 24.352. Use [...], Part, First, Last, and/or Take to extract the following from the list zeros. (a) The first and last elements, (b) the fourth through sixth elements, (c) the first three elements, and the last two elements. (d) Use Position to determine if and/or where 18.071 occurs in the list.
3. The **Fibonacci sequence** is defined by $f_0 = 1$, $f_1 = 1$, and $f_n = f_{n-1} + f_{n-2}$. (a) Define a Fibonacci function $f_n = f(n)$ that “remembers” the values computed. (b) Use your Fibonacci function to compute the first 15 values of the Fibonacci sequence. (c) Check that your results are correct by using Table and Fibonacci to compute the first 15 values of the Fibonacci sequence.
4. **(Mathematics of Finance)**
 - (a) **(Compound Interest)** A common problem in economics is the determination of the amount of interest earned from an investment. If P dollars are invested for t years at an annual interest rate of $r\%$ compounded m times per year, the **compound amount**, $A(t)$, at time t is given by

$$A(t) = P \left(1 + \frac{r}{m} \right)^{mt}.$$

If P dollars are invested for t years at an annual interest rate of $r\%$ compounded continuously, the compound amount, $A(t)$, at time t is given by $A(t) = Pe^{rt}$.

Suppose \$12,500 is invested at an annual rate of 7% compounded daily. How much money has accumulated and how much interest has been earned at the end of each 5-year period for $t = 0, 5, 10, 15, 20, 25, 30$? How much money has accumulated if interest is compounded continuously instead of daily?

- (b) Suppose \$10,000 is invested at an interest rate of 12% compounded daily. Create a table consisting of the total value of the investment and the interest earned at the end of 0, 5, 10, 15, 20, and 25 years.

What is the total value and interest earned on an investment of \$15,000 invested at an interest rate of 15% compounded daily at the end of 0, 10, 20, and 30 years?

- (c) **(Future Value)** If R dollars are deposited at the end of each period for n periods in an annuity that earns interest at a rate of $j\%$ per period, the **future value** of the annuity is

$$S_{\text{future}} = R \frac{(1+j)^n - 1}{j}.$$

Define a function `future` that calculates the future value of an annuity. Compute the future value of an annuity where \$250 is deposited at the end of each month for 60 months at a rate of 7% per year. Make a table of the future values of the annuity where \$150 is deposited at the end of each month for $12t$ months at a rate of 8% per year for $t = 1, 5, 9, 13, \dots, 21, 25$.

- (d) **(Annuity Due)** If R dollars are deposited at the beginning of each period for n periods with an interest rate of $j\%$ per period, the **annuity due** is

$$S_{\text{due}} = R \left[\frac{(1+j)^{n+1} - 1}{j} - 1 \right].$$

Define a function `due` that computes the annuity due. Use `due` to (a) compute the annuity due of \$500 deposited at the beginning of each month at an annual rate of 12% compounded monthly for 3 years, and (b) calculate the annuity due of \$100*k* deposited at the beginning of each month at an annual rate of 9% compounded monthly for 10 years for $k = 1, 2, 3, \dots, 10$.

Compare the annuity due on a \$100*k* monthly investment at an annual rate of 8% compounded monthly for $t = 5, 10, 15, 20$ and $k = 1, 2, 3, 4, 5$.

- (e) **(Present Value)** Another type of problem deals with determining the amount of money that must be invested in order to ensure a

particular return on the investment over a certain period of time. The **present value**, P , of an annuity of n payments of R dollars each at the end of consecutive interest periods with interest compounded at a rate of $j\%$ per period is

$$P = R \frac{1 - (1 + j)^{-n}}{j}.$$

Define a function `present` to compute the present value of an annuity. (a) Find the amount of money that would have to be invested at $7\frac{1}{2}\%$ compounded annually to provide an ordinary annuity income of \$45,000 per year for 40 years; and (b) find the amount of money that would have to be invested at 8% compounded annually to provide an ordinary annuity income of \$20,000 + \$5000 k per year for 35 years for $k = 0, 1, 2, 3, 4$, and 5 years.

- (f) (**Deferred Annuities**) The present value of a **deferred annuity** of R dollars per period for n periods deferred for k periods with an interest rate of j per period is

$$P_{def} = R \left[\frac{1 - (1 + j)^{-(n+k)}}{j} - \frac{1 - (1 + j)^{-k}}{j} \right].$$

Define a function `def[r,n,k,j]` to compute the value of a deferred annuity where r equals the amount of the deferred annuity, n equals the number of years in which the annuity is received, k equals the number of years in which the lump sum investment is made, and j equals the rate of interest. Use `def` to compute the lump sum that would have to be invested for 30 years at a rate of 15% compounded annually to provide an ordinary annuity income of \$35,000 per year for 35 years. How much money would have to be invested at the ages of 25, 35, 45, 55, and 65 at a rate of $8\frac{1}{2}\%$ compounded annually to provide an ordinary annuity income of \$30,000 per year for 40 years beginning at age 65?

- (g) (**Amortization**) A loan is **amortized** if both the principal and interest are paid by a sequence of equal periodic payments. A loan of P dollars at interest rate j per period may be amortized in n equal periodic payments of R dollars made at the end of each period, where

$$R = \frac{Pj}{1 - (1 + j)^{-n}}.$$

What is the monthly payment necessary to amortize a loan of \$75,000 with an interest rate of 9.5% compounded monthly over 20 years?

What is the monthly payment necessary to amortize a loan of \$80,000 at an annual rate of $j\%$ in 20 years for $j = 8, 8.5, 9, 9.5, 10$, and 10.5?

In many cases, the amount paid toward the principal of the loan and the total amount that remains to be paid after a certain payment need to be computed.

What is the unpaid balance of the principal at the end of the fifth year of a loan of \$60,000 with an annual interest rate of 8% scheduled to be amortized with monthly payments over a period of 10 years? What is the total interest paid immediately after the 60th payment?

- (h) What is the total interest paid on a loan of \$60,000 with an interest rate of 8% compounded monthly amortized over a period of 10 years (120 months) immediately after the 60th payment?
- (i) What is the monthly payment necessary to amortize a loan of \$45,000 with an interest rate of 7% compounded monthly over a period of 15 years (180 months)? What is the total principal and interest paid after 0, 3, 6, 9, 12, and 15 years?
- (j) Suppose that a loan of \$45,000 with interest rate of 7% compounded monthly is amortized over a period of 15 years (180 months). What is the principal and interest paid during each of the first 5 years of the loan?
- (k) *Challenge:* Suppose a retiree has \$1,200,000. If she can invest this sum at 7%, compounded annually, what level payment can she withdraw annually for a period of 40 years?
- (l) *Challenge:* Suppose an investor begins investing at a rate of d dollars per year at an annual rate of $j\%$. Each year the investor increases the amount invested by $i\%$. How much has the investor accumulated after m years?
- (m) Another interesting investment problem is discussed in the following exercise. In this case, Mathematica is useful in solving a recurrence equation that occurs in the problem. The command

$$\text{RSolve}[\{\text{equations}\}, a[n], n]$$

attempts to solve the recurrence equations equations for the variable $a[n]$ with no dependence on $a[j]$, $j \leq n - 1$.

I am 50 years old and I have \$500,000 that I can invest at a rate of 7% annually. Furthermore, I wish to receive a payment of \$50,000 the first year. Future annual payments should include cost-of-living adjustments at a rate of 3% annually. Is \$500,000 enough to guarantee this amount of annual income if I live to be 80 years old?

- (n) A 30-year mortgage of \$80,000 with an annual interest rate of 8.125% requires monthly payments of approximately \$600 (\$7200 annually) to amortize the loan in 30 years. However, using annuitytable, show that if the amount of the payments is increased by 3% each year, the 30-year mortgage is amortized in 17 years.
5. Define list to be a list of the first 100 positive integers. (a) Find the sum of the first 100 positive integers using Apply together with Plus. (b) Find the product of the first 100 positive integers using Apply and Product. (c) Describe the functionality of Apply. What is an abbreviated form?
6. Use RealDigits to find the first 101 digits in the decimal expansion of π . Use Table together with Count to determine the number of occurrences of each digit (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). *Challenge:* Repeat the exercise for a greater number of digits. Can you make a reasonable conclusion about the occurrence of each digit in the decimal expansion of π ?
7. Recall that a sequence of the form $x_{n+1} = f(x_n)$ is called a **dynamical system**.
- (a) Using $f(x) = x^2$ with $x_1 = a$, determine if $x_{n+1} = f(x_n)$ has a limit if $a = 1$, $a = 1.05$, and $a = 0.95$.

This dynamical system is said to have a **fixed point** at x if $f(x) = x$.

To find the fixed points $x_{n+1} = f(x_n)$ with $f(x) = x^2$, we solve $x^2 = x$ or $x^2 - x = 0$ with solutions and $x = 0$ and $x = 1$. In simple terms, a fixed point is called **stable** if a sequence that starts close to the fixed point has the fixed point as a limit. Otherwise, the fixed point is called **unstable**.

- (b) Would you classify $x = 1$ as stable or unstable? Would you classify $x = 0$ as stable or unstable? Briefly explain.
- (c) Consider $x_{n+1} = f(x_n)$ with $f(x) = 2x(1 - x)$.
- i. Find the two fixed points.
 - ii. Let $x_1 = 0.25$. Does the sequence $x_{n+1} = f(x_n)$ converge in this case? If so, what is the limit?
 - iii. Let $x_1 = 0.75$. Does the sequence $x_{n+1} = f(x_n)$ converge in this case? If so, what is the limit?
 - iv. Select any value of x_1 between 0 and 1. Does this choice affect the limit?
 - v. Classify the two fixed points as stable or unstable.
- (d) Sometimes, unusual behavior can be observed when working with dynamical systems. For example, consider the dynamical

system with $f(x) = x + 2.5x(1 - x)$ and $x_1 = 1.2$. We see that the sequence oscillates between 0.6 and 1.2. We say that the dynamical system has a **2-cycle** because the values of the sequence oscillate between two numbers.

- (e) Describe the behavior of $x_{n+1} = f(x_n)$ if $f(x) = x + 2.5x(1 - x)$ and $x_1 = 1.201$. Do you see a cycle? If so, how many numbers. What are these numbers? Does a small change in the initial value of the sequence affect the resulting values of the sequence based on the results of this problem and the previous example?
- (f) Describe the behavior of $x_{n+1} = f(x_n)$ if $f(x) = x + 2.5x(1 - x)$ and $x_1 = 1.3$. Do you see a cycle? If so, how many numbers. What are these numbers?
- (g) Describe the behavior of $x_{n+1} = f(x_n)$ if $f(x) = x + 2.5x(1 - x)$ and $x_1 = 1.2$. If the values do not seem to approach a single value or a cycle of several values, we say that the dynamical system is chaotic. Does this system appear to be chaotic?

In addition to your explanations, turn in the graphs obtained with plot for each problem.

8. Plot the Julia set for $f(z) = .36e^z$. *Hint:* Use the rectangle $a + bi$ for $0 \leq a \leq 5$ and $-2.5 \leq b \leq 2.5$.

Matrices and Vectors: Topics from Linear Algebra and Vector Calculus

Chapter 5 discusses operations on matrices and vectors, including topics from linear algebra, linear programming, and vector calculus.

5.1 NESTED LISTS: INTRODUCTION TO MATRICES, VECTORS, AND MATRIX OPERATIONS

5.1.1 Defining Nested Lists, Matrices, and Vectors

In Mathematica, a **matrix** is a list of lists where each list represents a row of the matrix. Therefore, the $m \times n$ matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{pmatrix}$$

is entered with

$$\mathbf{A} = \{\{a_{11}, a_{12}, \dots, a_{1n}\}, \{a_{21}, a_{22}, \dots, a_{2n}\}, \dots, \{a_{m1}, a_{m2}, \dots, a_{mn}\}\}$$

For example, to use Mathematica to define $\mathbf{m}$ to be the matrix $\mathbf{A} =$

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

enter the command

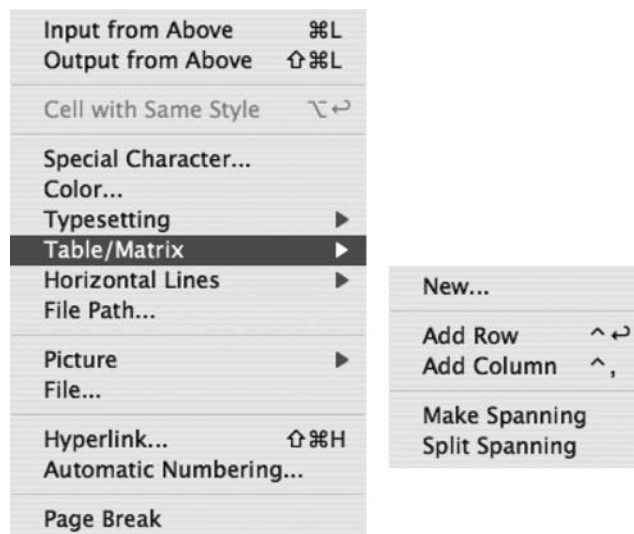
$$\mathbf{m} = \{\{a_{11}, a_{12}\}, \{a_{21}, a_{22}\}\}$$

The command `m=Array[a,{2,2}]` produces a result equivalent to this. Once a matrix $\mathbf{A}$ has been entered, it can be viewed in the traditional

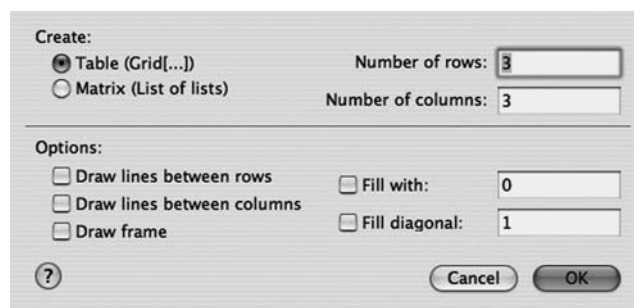
As when using `TableForm`, the result of using `MatrixForm` is no longer a list that can be manipulated using Mathematica commands. Use `MatrixForm` to view a matrix in traditional row-and-column form. Do not attempt to perform matrix operations on a `MatrixForm` object.

row-and-column form using the command `MatrixForm[A]`. You can quickly construct 2×2 matrices by clicking on the  button from the **BasicMathInput** palette, which is accessed by going to **Palettes** followed by **BasicMathInput**.

Alternatively, you can construct matrices of any dimension by going to the Mathematica menu under **Input** and selecting **Create Table/Matrix/Palette...**



The resulting pop-up window allows you to create tables, matrices, and palettes. To create a matrix, select **Matrix**, enter the number of rows and columns of the matrix, and select any other options. Pressing the **OK** button places the desired matrix at the position of the cursor in the Mathematica notebook.



Example 5.1.1 Use Mathematica to define the matrices $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ and $\begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{pmatrix}$.

Solution In this case, both `Table[aij, {i, 1, 3}, {j, 1, 3}]` and `Array[a, {3, 3}]` produce equivalent results when we define `matrixa` to be the matrix

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

The commands `MatrixForm` or `TableForm` are used to display the results in traditional matrix form.

Clear[a, b, matrixa, matrixb]

matrixa = Table[a_{ij}, {i, 1, 3}, {j, 1, 3}]

{ {a_{1,1}, a_{1,2}, a_{1,3}}, {a_{2,1}, a_{2,2}, a_{2,3}}, {a_{3,1}, a_{3,2}, a_{3,3}}}

MatrixForm[matrixa]

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

matrixa = Array[a, {3, 3}]

{ {a[1, 1], a[1, 2], a[1, 3]},

{a[2, 1], a[2, 2], a[2, 3]}, {a[3, 1], a[3, 2], a[3, 3]}}

MatrixForm[matrixa]

$$\begin{pmatrix} a[1, 1] & a[1, 2] & a[1, 3] \\ a[2, 1] & a[2, 2] & a[2, 3] \\ a[3, 1] & a[3, 2] & a[3, 3] \end{pmatrix}$$

We may also use Mathematica to define non-square matrices.

matrixb = Array[b, {2, 4}]

{ {b[1, 1], b[1, 2], b[1, 3], b[1, 4]}, {b[2, 1], b[2, 2], b[2, 3], b[2, 4]}}

MatrixForm[matrixb]

$$\begin{pmatrix} b[1, 1] & b[1, 2] & b[1, 3] & b[1, 4] \\ b[2, 1] & b[2, 2] & b[2, 3] & b[2, 4] \end{pmatrix}$$

Equivalent results would have been obtained by entering `Table[bij, {i, 1, 2}, {j, 1, 4}]`.

More generally, the commands `Table[f[i, j], {i, imax}, {j, jmax}]` and `Array[f, {imax, jmax}]` yield nested lists corresponding to the $\text{imax} \times \text{jmax}$ matrix

$$\begin{pmatrix} f(1, 1) & f(1, 2) & \cdots & f(1, \text{jmax}) \\ f(2, 1) & f(2, 2) & \cdots & f(2, \text{jmax}) \\ \vdots & \vdots & \ddots & \vdots \\ f(\text{imax}, 1) & f(\text{imax}, 2) & \cdots & f(\text{imax}, \text{jmax}) \end{pmatrix}.$$

`Table[f[i, j], {i, imin, imax, istep}, {j, jmin, jmax, jstep}]` returns the list of lists

```

{{f[imin, jmin], f[imin, jmin + jstep], ..., f[imin, jmax]},
 {f[imin + istep, jmin], ..., f[imin + istep, jmax]},
 ..., {f[imax, jmin], ..., f[imax, jmax]}}

```

and the command

```

Table[f[i, j, k, ...], {i, imin, imax, istep}, {j, jmin, jmax, jstep},
 {k, kmin, kmax, kstep}, ...]

```

calculates a nested list; the list associated with i is outermost. If `istep` is omitted, the step size is one.

Example 5.1.2 Define $\mathbf{C}$ to be the 3×4 matrix (c_{ij}) , where c_{ij} , the entry in the i th row and j th column of $\mathbf{C}$, is the numerical value of $\cos(j^2 - i^2) \sin(i^2 - j^2)$.

Solution After clearing all prior definitions of `c`, if any, we define `c[i, j]` to be the numerical value of $\cos(j^2 - i^2) \sin(i^2 - j^2)$ and then use `Array` to compute the 3×4 matrix `matrixc`.

```

Clear[c, matrixc]
c[i_, j_] = N[Cos[j^2 - i^2] Sin[i^2 - j^2]]
Cos[i^2 - 1. j^2] Sin[i^2 - 1. j^2]

matrixc = Array[c, {3, 4}]
{0., 0.139708, 0.143952, 0.494016},
{-0.139708, 0., 0.272011, 0.452789},
{-0.143952, -0.272011, 0., -0.495304}

```

```

MatrixForm[matrixc]

```

$$\begin{pmatrix} 0. & 0.139708 & 0.143952 & 0.494016 \\ -0.139708 & 0. & 0.272011 & 0.452789 \\ -0.143952 & -0.272011 & 0. & -0.495304 \end{pmatrix}$$

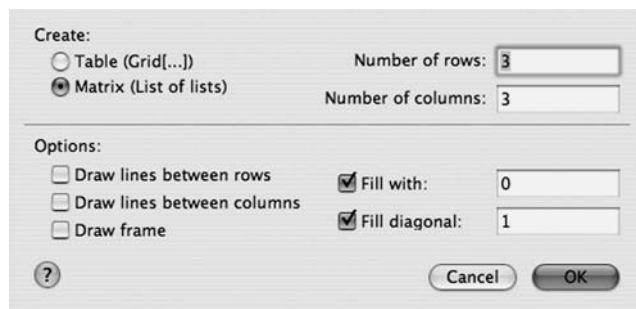
Example 5.1.3 Define the matrix $\mathbf{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

Solution The matrix $\mathbf{I}_3$ is the 3×3 **identity matrix**. Generally, the $n \times n$ matrix with 1's on the diagonal and 0's elsewhere is the $n \times n$ identity matrix. The command `IdentityMatrix[n]` returns the $n \times n$ identity matrix.

IdentityMatrix[3]

`{{1,0,0}, {0,1,0}, {0,0,1}}`

The same result is obtained by going to **Insert** under the Mathematica menu and selecting **Table/Matrix/** followed by **New...** We then check **Matrix**, **Fill with:** 0 and **Fill diagonal:** 1.



Pressing the **OK** button inserts the 3×3 identity matrix at the location of the cursor.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

`{{1,0,0}, {0,1,0}, {0,0,1}}`

With Mathematica, you do not need to distinguish between row and column vectors. Provided that computations are well-defined, Mathematica carries them out correctly. Mathematica warns of any ambiguities when they (rarely) occur.

In Mathematica, a **vector** is a list of numbers and, thus, is entered in the same manner as lists. For example, to use Mathematica to define the row vector `vectorv` to be $(v_1 \ v_2 \ v_3)$, enter `vectorv={v1,v2,v3}`. Similarly, to define the column vector `vectorv` to be $\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$, enter `vectorv={v1,v2,v3}` or `vectorv={{v1},{v2},{v3}}`.

Generally, with Mathematica you do not need to distinguish between row and column vectors: Mathematica usually performs computations with vectors and matrices correctly as long as the computations are well-defined.

Example 5.1.4 Define the vector $\mathbf{w} = \begin{pmatrix} -4 \\ -5 \\ 2 \end{pmatrix}$, `vectorv` to be the vector $(v_1 \ v_2 \ v_3 \ v_4)$ and `zerovec` to be the vector $(0 \ 0 \ 0 \ 0 \ 0)$.

Solution To define **w**, we enter

```
w = {-4, -5, 2}
{-4, -5, 2}
```

or

```
w = {{-4}, {-5}, {2}};
MatrixForm[w]

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

```

To define **vectorv**, we use **Array**.

```
vectorv = Array[v, 4]
{v[1], v[2], v[3], v[4]}
```

Equivalent results would have been obtained by entering **Table[v, {i, 1, 4}]**. To define **zerovec**, we use **Table**.

```
zerovec = Table[0, {5}]
{0, 0, 0, 0, 0}
```

The same result is obtained by going to **Insert** under the Mathematica menu and selecting **Table/Matrix** to create the zero vector.

```
{0 0 0 0 0}
{{0, 0, 0, 0, 0}}
```

5.1.2 Extracting Elements of Matrices

For the 2×2 matrix $m = \{\{a_{1,1}, a_{1,2}\}, \{a_{2,1}, a_{2,2}\}\}$ defined previously, **m[[1]]** yields the first element of matrix **m** which is the list $\{a_{1,1}, a_{1,2}\}$ or the first row of **m**; **m[[2,1]]** yields the first element of the second element of matrix **m** which is $a_{2,1}$. In general, if **m** is an $i \times j$ matrix, **m[[i,j]]** or **Part[m,i,j]** returns the unique element in the i th row and j th column of **m**. Specifically, **m[[i,j]]** yields the j th part of the i th part of **m**; **list[[i]]** or **Part[list,i]** yields the i th part of **list**; **list[[i,j]]** or **Part[list,i,j]** yields the j th part of the i th part of **list**, and so on.

Example 5.1.5 Define **mb** to be the matrix $\begin{pmatrix} 10 & -6 & -9 \\ 6 & -5 & -7 \\ -10 & 9 & 12 \end{pmatrix}$. (a) Extract the third row of **mb**.
 (b) Extract the element in the first row and third column of **mb**. (c) Display **mb** in traditional matrix form.

Solution We begin by defining `mb`. `mb[[i,j]]` yields the (unique) number in the i th row and j th column of `mb`. Observe how various components of `mb` (rows and elements) can be extracted and how `mb` is placed in `MatrixForm`.

```
mb = {{10, -6, -9}, {6, -5, -7}, {-10, 9, 12}};
```

```
MatrixForm[mb]
```

$$\begin{pmatrix} 10 & -6 & -9 \\ 6 & -5 & -7 \\ -10 & 9 & 12 \end{pmatrix}$$

```
mb[[3]]
```

```
{-10, 9, 12}
```

```
mb[[1, 3]]
```

```
-9
```

If `m` is a matrix, the i th row of `m` is extracted with `m[[i]]`. The command `Transpose[m]` yields the transpose of the matrix `m`, the matrix obtained by interchanging the rows and columns of `m`. We extract columns of `m` by computing `Transpose[m]` and then using `Part` to extract rows from the transpose. Namely, if `m` is a matrix, `Transpose[m] [[i]]` extracts the i th row from the transpose of `m` which is the same as the i th column of `m`.

Alternatively, if `A` is $n \times m$ (rows $\times$ columns), the i th column of `A` is the vector that consists of the i th part of each row of the matrix, so given an i -value `Table[A[[j,i]],{j,1,n}]` returns the i th column of `A`.

Example 5.1.6 Extract the second and third columns from $\mathbf{A} = \begin{pmatrix} 0 & -2 & 2 \\ -1 & 1 & -3 \\ 2 & -4 & 1 \end{pmatrix}$.

Solution We first define `matrixa` and then use `Transpose` to compute the transpose of `matrixa`, naming the result `ta`, and then displaying `ta` in `MatrixForm`.

```
matrixa = {{0, -2, 2}, {-1, 1, -3}, {2, -4, 1}};
```

```
MatrixForm[matrixa]
```

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

```
ta = Transpose[matrixa]
```

```
MatrixForm[ta]
```

```
{{0, -1, 2}, {-2, 1, -4}, {2, -3, 1}}
```

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

Next, we extract the second column of `matrixa` using `Transpose` together with `Part` (`[[...]]`). Because we have already defined `ta` to be the transpose of `matrixa`, entering `ta[[2]]` would produce the same result.

Transpose[matrixa][[2]]

`{-2, 1, -4}`

To extract the third column, we take advantage of the fact that we have already defined `ta` to be the transpose of `matrixa`. Entering `Transpose[matrixa][[3]]` produces the same result.

ta[[3]]

`{2, -3, 1}`

You can also use `Take` to extract elements of lists and matrices. Entering

Take[matrixa, 2]

Take[matrixa, 2]//MatrixForm

`{{0, -2, 2}, {-1, 1, -3}}`

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

returns the first two rows of `matrixa` because the first two parts of `matrixa` are the lists corresponding to those rows. Similarly,

Take[matrixa, {2}]

Take[matrixa, {2}]//MatrixForm

`{{-1, 1, -3}}`

$$\begin{pmatrix} -1 & 1 & -3 \end{pmatrix}$$

returns the second row, whereas

Take[matrixa, {2, 3}]

Take[matrixa, {2, 3}]//MatrixForm

`{{-1, 1, -3}, {2, -4, 1}}`

$$\begin{pmatrix} -1 & 1 & -3 \\ 2 & -4 & 1 \end{pmatrix}$$

returns the second and third rows.

The example illustrates that `Take[list, n]` returns the first n elements of `list`; `Take[list, {n}]` returns the n th element of `list`; `Take[list, {n1, n2, ...}]` returns the n_1 st, n_2 nd, ... elements of `list`; and so on.

5.1.3 Basic Computations with Matrices

Mathematica performs all of the usual operations on matrices. Matrix addition ($A + B$), scalar multiplication (kA), matrix multiplication (when defined) (AB), and combinations of these operations are all possible. The **transpose** of A , A^t , is obtained by interchanging the rows and columns of A and is computed with the command `Transpose[A]`. If A is a square matrix, the determinant of A is obtained with `Det[A]`.

If A and B are $n \times n$ matrices satisfying $AB = BA = I$, where I is the $n \times n$ matrix with 1's on the diagonal and 0's elsewhere (the $n \times n$ identity matrix), B is called the **inverse** of A and is denoted by A^{-1} . If the inverse of a matrix A exists, the inverse is found with `Inverse[A]`. Thus, assuming that $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ has an inverse ($ad - bc \neq 0$), the inverse is

Inverse[[{a, b}, {c, d}]]

$$\left\{ \left\{ \frac{d}{-bc+ad}, -\frac{b}{-bc+ad} \right\}, \left\{ -\frac{c}{-bc+ad}, \frac{a}{-bc+ad} \right\} \right\}$$

Example 5.1.7 Let $A = \begin{pmatrix} 3 & -4 & 5 \\ 8 & 0 & -3 \\ 5 & 2 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 10 & -6 & -9 \\ 6 & -5 & -7 \\ -10 & 9 & 12 \end{pmatrix}$. Compute (a) $A + B$; (b) $B - 4A$; (c) the inverse of AB ; (d) the transpose of $(A - 2B)B$; and (e) $\det A = |A|$.

Solution We enter **ma** (corresponding to **A**) and **mb** (corresponding to **B**) as nested lists, where each element corresponds to a row of the matrix. We suppress the output by ending each command with a semicolon.

ma = {{3, -4, 5}, {8, 0, -3}, {5, 2, 1}};

mb = {{10, -6, -9}, {6, -5, -7}, {-10, 9, 12}};

Entering

ma + mb//MatrixForm

$$\begin{pmatrix} 13 & -10 & -4 \\ 14 & -5 & -10 \\ -5 & 11 & 13 \end{pmatrix}$$

adds matrix **ma** to **mb** and expresses the result in traditional matrix form.

Entering

mb - 4ma//MatrixForm

$$\begin{pmatrix} -2 & 10 & -29 \\ -26 & -5 & 5 \\ -30 & 1 & 8 \end{pmatrix}$$

subtracts four times matrix **ma** from **mb** and expresses the result in traditional matrix form. Entering

Inverse[ma·mb]//MatrixForm

$$\begin{pmatrix} \frac{59}{380} & \frac{53}{190} & -\frac{167}{380} \\ -\frac{223}{570} & -\frac{92}{95} & \frac{979}{570} \\ \frac{49}{114} & \frac{18}{19} & -\frac{187}{114} \end{pmatrix}$$

Matrix products, when defined, are computed by placing a period (.) between the matrices being multiplied. Note that a period is also used to compute the dot product of two vectors, when the dot product is defined.

computes the inverse of the matrix product **AB**. Similarly, entering

Transpose[(ma-2mb)·mb]//MatrixForm

$$\begin{pmatrix} -352 & -90 & 384 \\ 269 & 73 & -277 \\ 373 & 98 & -389 \end{pmatrix}$$

computes the transpose of **(A - 2B)B**, and entering

Det[ma]

190

computes the determinant of **A**.

Example 5.1.8 Compute **AB** and **BA** if $\mathbf{A} = \begin{pmatrix} -1 & -5 & -5 & -4 \\ -3 & 5 & 3 & -2 \\ -4 & 4 & 2 & -3 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 1 & -2 \\ -4 & 3 \\ 4 & -4 \\ -5 & -3 \end{pmatrix}$.

Solution Because **A** is a 3×4 matrix and **B** is a 4×2 matrix, **AB** is defined and is a 3×2 matrix. We define **matrixa** and **matrixb** with the following commands.

Remember that you can also define matrices by going to **Insert** under the **Mathematica** menu and selecting **Table/Matrix**. After entering the desired number of rows and columns and pressing the **OK** button, a matrix template is placed at the location of the cursor that you can fill in.

$$\mathbf{matrixa} = \begin{pmatrix} -1 & -5 & -5 & -4 \\ -3 & 5 & 3 & -2 \\ -4 & 4 & 2 & -3 \end{pmatrix};$$

$$\mathbf{matrixb} = \begin{pmatrix} 1 & -2 \\ -4 & 3 \\ 4 & -4 \\ -5 & -3 \end{pmatrix};$$

We then compute the product, naming the result **ab**, and display **ab** in **MatrixForm**.

ab = matrixa.matrixb;

MatrixForm[ab]

$$\begin{pmatrix} 19 & 19 \\ -1 & 15 \\ 3 & 21 \end{pmatrix}$$

However, the matrix product $\mathbf{BA}$ is not defined and Mathematica produces error messages when we attempt to compute it.

```

matrixa =  $\begin{pmatrix} -1 & -5 & -5 & -4 \\ -3 & 5 & 3 & -2 \\ -4 & 4 & 2 & -3 \end{pmatrix}$ ;

matrixb =  $\begin{pmatrix} 1 & -2 \\ -4 & 3 \\ 4 & -4 \\ -5 & -3 \end{pmatrix}$ ;

ab = matrixa.matrixb;
MatrixForm[ab]

 $\begin{pmatrix} 19 & 19 \\ -1 & 15 \\ 3 & 21 \end{pmatrix}$ 

matrixb.matrixa
Dot::dotsh : Tensors  $\{\{1, -2\}, \{-4, 3\}, \{4, -4\}, \{-5, -3\}\}$ 
and  $\{\{-1, -5, -5, -4\}, \{-3, 5, 3, -2\}, \{-4, 4, 2, -3\}\}$ 
have incompatible shapes. >>

 $\{\{1, -2\}, \{-4, 3\}, \{4, -4\}, \{-5, -3\}\}.$ 

 $\{\{-1, -5, -5, -4\}, \{-3, 5, 3, -2\}, \{-4, 4, 2, -3\}\}$ 

```

Special attention must be given to the notation that must be used in taking the product of a square matrix with itself. The following example illustrates how Mathematica interprets the expression $(\text{matrixb})^n$. The command $(\text{matrixb})^n$ raises each element of the matrix matrixb to the n th power. The command `MatrixPower` is used to compute powers of matrices.

Example 5.1.9 Let $\mathbf{B} = \begin{pmatrix} -2 & 3 & 4 & 0 \\ -2 & 0 & 1 & 3 \\ -1 & 4 & -6 & 5 \\ 4 & 8 & 11 & -4 \end{pmatrix}$. (a) Compute $\mathbf{B}^2$ and $\mathbf{B}^3$. (b) Cube each entry of $\mathbf{B}$.

Solution After defining $\mathbf{B}$, we compute $\mathbf{B}^2$. The same results would have been obtained by entering `MatrixPower[matrixb,2]`.

```

matrixb = {{-2, 3, 4, 0}, {-2, 0, 1, 3}, {-1, 4, -6, 5},
{4, 8, 11, -4}};
MatrixForm[matrixb . matrixb]

 $\begin{pmatrix} -6 & 10 & -29 & 29 \\ 15 & 22 & 19 & -7 \\ 20 & 13 & 91 & -38 \\ -51 & 24 & -86 & 95 \end{pmatrix}$ 

```

Next, we use `MatrixPower` to compute $\mathbf{B}^3$. The same results would be obtained by entering `matrixb.matrixb.matrixb`.

MatrixForm[MatrixPower[matrixb, 3]]

$$\begin{pmatrix} 137 & 98 & 479 & -231 \\ -121 & 65 & -109 & 189 \\ -309 & 120 & -871 & 646 \\ 520 & 263 & 1381 & -738 \end{pmatrix}$$

Last, we cube each entry of $\mathbf{B}$ with $\wedge$.

MatrixForm[matrixb³]

$$\begin{pmatrix} -8 & 27 & 64 & 0 \\ -8 & 0 & 1 & 27 \\ -1 & 64 & -216 & 125 \\ 64 & 512 & 1331 & -64 \end{pmatrix}$$

If $|\mathbf{A}| \neq 0$, the inverse of $\mathbf{A}$ can be computed using the formula

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \mathbf{A}^a, \quad (5.1)$$

The **cofactor matrix**, $\mathbf{A}^c$, of $\mathbf{A}$ is the matrix obtained by replacing each element of $\mathbf{A}$ by its cofactor.

where $\mathbf{A}^a$ is the *transpose of the cofactor matrix*.

If $\mathbf{A}$ has an inverse, reducing the matrix $(\mathbf{A}|\mathbf{I})$ to reduced row echelon form results in $(\mathbf{I}|\mathbf{A}^{-1})$. This method is often easier to implement than (5.1).

Example 5.1.10 Calculate $\mathbf{A}^{-1}$ if $\mathbf{A} = \begin{pmatrix} 2 & -2 & 1 \\ 0 & -2 & 2 \\ -2 & -1 & -1 \end{pmatrix}$.

Solution After defining $\mathbf{A}$ and $\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, we compute $|\mathbf{A}| = 12$, so $\mathbf{A}^{-1}$ exists.

capa = {{2, -2, 1}, {0, -2, 2}, {-2, -1, -1}}

i3 = IdentityMatrix[3]

{{2, -2, 1}, {0, -2, 2}, {-2, -1, -1}}

{{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}

Det[capa]

12

`Join[a,b,n]` concatenates lists $\mathbf{a}$ and $\mathbf{b}$ at level n . For matrices the level one objects (`capa[[i]]`) are the rows; the level two objects (`capa[[i,j]]`) are the entries.

Thus, `Join[capa,i3]` returns the matrix $\begin{pmatrix} \mathbf{A} \\ \mathbf{I} \end{pmatrix}$, whereas `Join[capa,i3,2]` forms the matrix $(\mathbf{A}|\mathbf{I})$.

```
ai3 = Join[capa, i3, 2]
{{2, -2, 1, 1, 0, 0}, {0, -2, 2, 0, 1, 0},
 {-2, -1, -1, 0, 0, 1}}
```

```
MatrixForm[ai3]
```

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

RowReduce[A]
reduces **A** to
reduced
row echelon
form.

We then use RowReduce to reduce $(\mathbf{A}|\mathbf{I})$ to row echelon form.

```
rrai3 = RowReduce[ai3]
```

```
{{1, 0, 0, 1/3, -1/4, -1/6}, {0, 1, 0, -1/3, 0, -1/3}, {0, 0, 1, -1/3, 1/2, -1/3}}
```

```
MatrixForm[rrai3]
```

$$\begin{pmatrix} 1 & 0 & 0 & \frac{1}{3} & -\frac{1}{4} & -\frac{1}{6} \\ 0 & 1 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & 0 & 1 & -\frac{1}{3} & \frac{1}{2} & -\frac{1}{3} \end{pmatrix}$$

The result indicates that $\mathbf{A}^{-1} = \begin{pmatrix} 1/3 & -1/4 & -1/6 \\ -1/3 & 0 & -1/3 \\ -1/3 & 1/2 & -1/3 \end{pmatrix}$.

5.1.4 Basic Computations with Vectors

Basic Operations on Vectors

Computations with vectors are performed in the same way as computations with matrices.

Example 5.1.11 Let $\mathbf{v} = \begin{pmatrix} 0 \\ 5 \\ 1 \\ 2 \end{pmatrix}$ and $\mathbf{w} = \begin{pmatrix} 3 \\ 0 \\ 4 \\ -2 \end{pmatrix}$. (a) Calculate $\mathbf{v} - 2\mathbf{w}$ and $\mathbf{v} \cdot \mathbf{w}$. (b) Find a unit vector

with the same direction as $\mathbf{v}$ and a unit vector with the same direction as $\mathbf{w}$.

Solution We begin by defining $\mathbf{v}$ and $\mathbf{w}$ and then compute $\mathbf{v} - 2\mathbf{w}$ and $\mathbf{v} \cdot \mathbf{w}$.

```
v = {0, 5, 1, 2};
```

```
w = {3, 0, 4, -2};
```

```
v - 2w
```

```
{-6, 5, -7, 6}
```

```
v . w
```

```
0
```

The **norm** of the vector $\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$ is

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2} = \sqrt{\mathbf{v} \cdot \mathbf{v}}.$$

The command **Norm[v]** returns the norm of the vector $\mathbf{v}$.

If k is a scalar, the direction of $k\mathbf{v}$ is the same as the direction of $\mathbf{v}$. Thus, if $\mathbf{v}$ is a nonzero vector, the vector $\frac{1}{\|\mathbf{v}\|}\mathbf{v}$ has the same direction as $\mathbf{v}$ and because

$$\left\| \frac{1}{\|\mathbf{v}\|}\mathbf{v} \right\| = \frac{1}{\|\mathbf{v}\|}\|\mathbf{v}\| = 1, \quad \frac{1}{\|\mathbf{v}\|}\mathbf{v} \text{ is a unit vector.}$$

First, we compute $\|\mathbf{v}\|$ with **Norm**. We then compute $\frac{1}{\|\mathbf{v}\|}\mathbf{v}$, calling the result $\mathbf{uv}$, and $\frac{1}{\|\mathbf{w}\|}\mathbf{w}$. The results correspond to unit vectors with the same direction as $\mathbf{v}$ and $\mathbf{w}$, respectively.

Norm[v]

$$\sqrt{30}$$

$$\mathbf{uv} = \frac{\mathbf{v}}{\text{Norm}[\mathbf{v}]}$$

$$\left\{ 0, \sqrt{\frac{5}{6}}, \frac{1}{\sqrt{30}}, \sqrt{\frac{2}{15}} \right\}$$

Norm[uv]

$$1$$

$$\frac{\mathbf{w}}{\text{Norm}[\mathbf{w}]}$$

$$\left\{ \frac{3}{\sqrt{29}}, 0, \frac{4}{\sqrt{29}}, -\frac{2}{\sqrt{29}} \right\}$$

Basic Operations on Vectors in 3-Space

We review the elementary properties of vectors in 3-space. Let

$$\mathbf{u} = \langle u_1, u_2, u_3 \rangle = u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}$$

and

$$\mathbf{v} = \langle v_1, v_2, v_3 \rangle = v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k}$$

be vectors in space.

1. $\mathbf{u}$ and $\mathbf{v}$ are **equal** if and only if their components are equal:

$$\mathbf{u} = \mathbf{v} \Leftrightarrow u_1 = v_1, u_2 = v_2, \text{ and } u_3 = v_3.$$

Vector calculus is discussed in Section 5.5.

In space, the **standard unit vectors** are $\mathbf{i} = \langle 1, 0, 0 \rangle$, $\mathbf{j} = \langle 0, 1, 0 \rangle$, and $\mathbf{k} = \langle 0, 0, 1 \rangle$. With the exception of the cross product, the vector operations discussed here are performed in the same way for vectors in the plane as they are in space. In the plane, the **standard unit vectors** are $\mathbf{i} = \langle 1, 0 \rangle$ and $\mathbf{j} = \langle 0, 1 \rangle$.

A **unit vector** is a vector with length 1.

2. The **length** (or **norm**) of $\mathbf{u}$ is

$$\|\mathbf{u}\| = \sqrt{u_1^2 + u_2^2 + u_3^2}.$$

3. If c is a scalar (number),

$$c\mathbf{u} = \langle cu_1, cu_2, cu_3 \rangle.$$

4. The **sum** of $\mathbf{u}$ and $\mathbf{v}$ is defined to be the vector

$$\mathbf{u} + \mathbf{v} = \langle u_1 + v_1, u_2 + v_2, u_3 + v_3 \rangle.$$

5. If $\mathbf{u} \neq \mathbf{0}$, a unit vector with the same direction as $\mathbf{u}$ is

$$\frac{1}{\|\mathbf{u}\|} \mathbf{u} = \frac{1}{\sqrt{u_1^2 + u_2^2 + u_3^2}} \langle u_1, u_2, u_3 \rangle.$$

6. $\mathbf{u}$ and $\mathbf{v}$ are **parallel** if there is a scalar c so that $\mathbf{u} = c\mathbf{v}$.

7. The **dot product** of $\mathbf{u}$ and $\mathbf{v}$ is

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + u_3v_3.$$

If θ is the angle between $\mathbf{u}$ and $\mathbf{v}$,

$$\cos \theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}.$$

Consequently, $\mathbf{u}$ and $\mathbf{v}$ are orthogonal if $\mathbf{u} \cdot \mathbf{v} = 0$.

8. The **cross product** of $\mathbf{u}$ and $\mathbf{v}$ is

$$\begin{aligned} \mathbf{u} \times \mathbf{v} &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} \\ &= (u_2v_3 - u_3v_2)\mathbf{i} - (u_1v_3 - u_3v_1)\mathbf{j} + (u_1v_2 - u_2v_1)\mathbf{k}. \end{aligned}$$

You should verify that $\mathbf{u} \cdot (\mathbf{u} \times \mathbf{v}) = 0$ and $\mathbf{v} \cdot (\mathbf{u} \times \mathbf{v}) = 0$. Hence, $\mathbf{u} \times \mathbf{v}$ is orthogonal to both $\mathbf{u}$ and $\mathbf{v}$.

Topics from linear algebra (including determinants) are discussed in more detail in the next sections. For now, we illustrate several of the basic

operations listed previously: $\mathbf{u} \cdot \mathbf{v}$ and $\text{Dot}[\mathbf{u}, \mathbf{v}]$ compute $\mathbf{u} \cdot \mathbf{v}$; $\text{Cross}[\mathbf{u}, \mathbf{v}]$ computes $\mathbf{u} \times \mathbf{v}$.

Example 5.1.12 Let $\mathbf{u} = \langle 3, 4, 1 \rangle$ and $\mathbf{v} = \langle -4, 3, -2 \rangle$. Calculate (a) $\mathbf{u} \cdot \mathbf{v}$, (b) $\mathbf{u} \times \mathbf{v}$, (c) $\|\mathbf{u}\|$, and (d) $\|\mathbf{v}\|$. (e) Find the angle between $\mathbf{u}$ and $\mathbf{v}$. (f) Find unit vectors with the same direction as $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{u} \times \mathbf{v}$.

Solution We begin by defining $\mathbf{u} = \langle 3, 4, 1 \rangle$ and $\mathbf{v} = \langle -4, 3, -2 \rangle$. Notice that to define $\mathbf{u} = \langle u_1, u_2, u_3 \rangle$ with Mathematica, we use the form

$$\mathbf{u} = \{\mathbf{u1}, \mathbf{u2}, \mathbf{u3}\}$$

Similarly, to define $\mathbf{u} = \langle u_1, u_2 \rangle$, we use the form $\mathbf{u} = \{\mathbf{u1}, \mathbf{u2}\}$.

We illustrate the use of Dot and Cross to calculate (a)–(d).

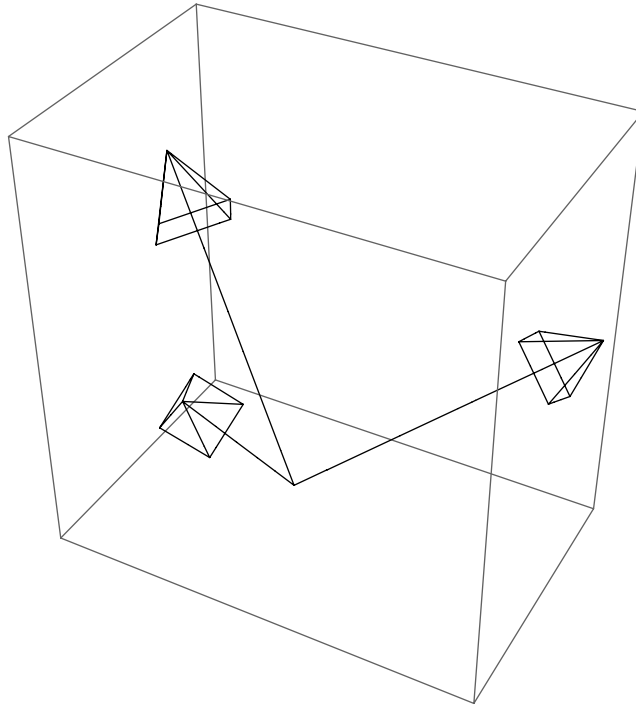
```
u = {3, 4, 1};
v = {-4, 3, -2};
udv = Dot[u, v]
-2
u · v
-2
ucv = Cross[u, v]
{-11, 2, 25}
nu = Norm[u]
√26
nv = Sqrt[v · v]
√29
```

We use the formula $\theta = \cos^{-1} \left(\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \right)$ to find the angle θ between $\mathbf{u}$ and $\mathbf{v}$.

```
ArcCos[u · v / (nu nv)]
N[%]
ArcCos  $\left[ -\sqrt{\frac{2}{377}} \right]$ 
1.6437
```

Unit vectors with the same direction as $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{u} \times \mathbf{v}$ are found next.

```
normu = u/nu
normv = v/nv
 $\left\{ \frac{3}{\sqrt{26}}, 2\sqrt{\frac{2}{13}}, \frac{1}{\sqrt{26}} \right\}$ 
 $\left\{ -\frac{4}{\sqrt{29}}, \frac{3}{\sqrt{29}}, -\frac{2}{\sqrt{29}} \right\}$ 
nucrossv = ucv/Norm[ucv]
 $\left\{ -\frac{11}{5\sqrt{30}}, \frac{\sqrt{\frac{2}{15}}}{5}, \sqrt{\frac{5}{6}} \right\}$ 
```

**FIGURE 5.1**

Orthogonal vectors

We can graphically confirm that these three vectors are orthogonal by graphing all three vectors with the **ListVectorFieldPlot3D** function, which is contained in the **VectorFieldPlots** package. After loading the **ListVectorFieldPlot3D** package, the command

ListVectorFieldPlot3D[listofvectors]

graphs the list of vectors **listofvectors**. Each element of **listofvectors** is of the form $\{\{u_1, u_2, u_3\}, \{v_1, v_2, v_3\}\}$, where (u_1, u_2, u_3) and (v_1, v_2, v_3) are the initial and terminal points of each vector. We show the vectors in Figure 5.1.

Needs["VectorFieldPlots"]

**ListVectorFieldPlot3D[{{{0, 0, 0}, normu},
 {{0, 0, 0}, normv}, {{0, 0, 0}, nucrossv}},
 VectorHeads → True]**

In the plot, the vectors do appear to be orthogonal as expected.

With the exception of the cross product, the calculations described previously can also be performed on vectors in the plane.

Example 5.1.13 If $\mathbf{u}$ and $\mathbf{v}$ are nonzero vectors, the **projection** of $\mathbf{u}$ onto $\mathbf{v}$ is

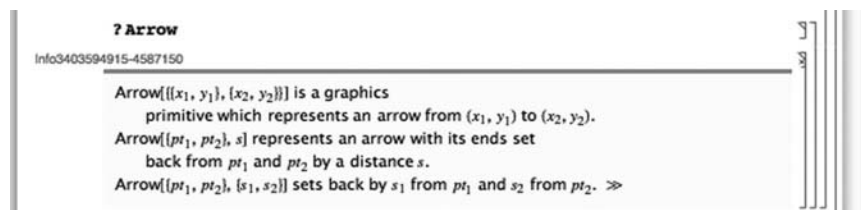
$$\text{proj}_{\mathbf{v}} \mathbf{u} = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{v}.$$

Find $\text{proj}_{\mathbf{v}} \mathbf{u}$ if $\mathbf{u} = \langle -1, 4 \rangle$ and $\mathbf{v} = \langle 2, 6 \rangle$.

Solution First, we define $\mathbf{u} = \langle -1, 4 \rangle$ and $\mathbf{v} = \langle 2, 6 \rangle$ and then compute $\text{proj}_{\mathbf{v}} \mathbf{u}$.

```
u = {-1, 4};
v = {2, 6};
projvu = u.v v/v.v
{ 11/10, 33/10 }
```

Next, we graph $\mathbf{u}$, $\mathbf{v}$, and $\text{proj}_{\mathbf{v}} \mathbf{u}$ together using `Arrow`, `Show`, and `GraphicsRow` in Figure 5.2.



```
p1 = Show[Graphics[{Arrowheads[Medium], Arrow[{{0, 0}, u]},
Arrow[{{0, 0}, v}],
Thickness[.05], Arrow[{{0, 0}, projvu}]}],
Axes -> Automatic, AspectRatio -> Automatic];
p2 = Show[Graphics[{Arrowheads[Medium], Arrow[{{0, 0}, u]},
```

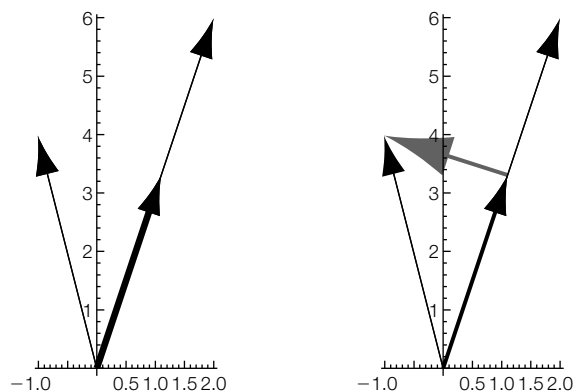
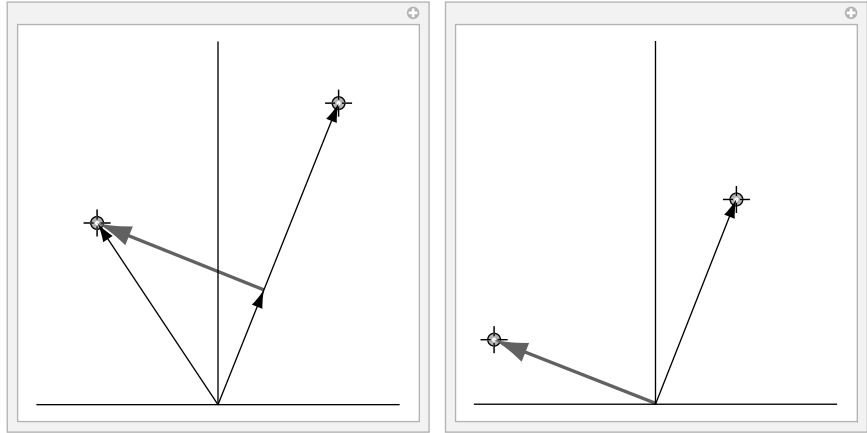


FIGURE 5.2

Projection of a vector

**FIGURE 5.3**

Using Manipulate to visualize the projection of one vector onto another

```

Arrow[{{0, 0}, v}],
Thickness[.03], Arrow[{{0, 0}, projvu}], GrayLevel[.4],
Arrowheads[Large], Arrow[{projvu, u}]],
Axes → Automatic, AspectRatio → Automatic];
Show[GraphicsRow[{p1, p2}]]

```

In the graph, notice that $\mathbf{u} = \text{proj}_v \mathbf{u} + (\mathbf{u} - \text{proj}_v \mathbf{u})$ and the vector $\mathbf{u} - \text{proj}_v \mathbf{u}$ is perpendicular to $\mathbf{v}$.

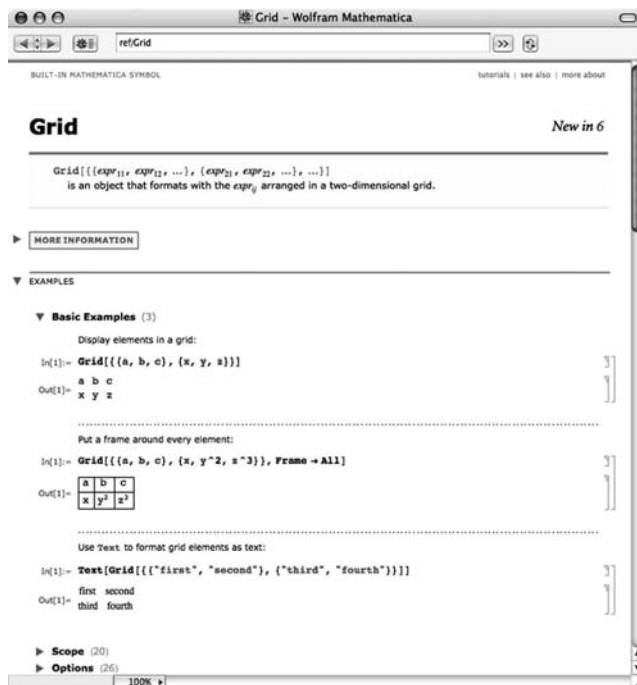
With the following, we use Manipulate to generalize the example. See Figure 5.3.

```

Clear[u, v, projvu, p1, p2];
Manipulate[
projvu = u.vv/v.v;
Show[Graphics[{Arrowheads[Medium], Arrow[{{0, 0}, u}],
Arrow[{{0, 0}, v}],
Thickness[.005], Arrow[{{0, 0}, projvu}], GrayLevel[.4],
Arrowheads[Large], Arrow[{projvu, u}]]},
Axes → Automatic, PlotRange → {{-3, 3}, {0, 6}},
AspectRatio → Automatic, Ticks → None], {{u, {-2, 3}}, Locator},
{{v, {2, 5}}, Locator}]

```

If you only need to display a two-dimensional array in row-and-column form, it is easier to use Grid rather than Table together with TableForm or MatrixForm.



For a list of all the options associated with Grid, enter Options[Grid].

```
Options[Grid]
{Alignment -> {Center, Baseline}, Background -> None,
 BaselinePosition -> Automatic, BaseStyle -> {},
 DefaultBaseStyle -> Grid, DefaultElement -> {},
 Dividers -> None, Frame -> None, FrameStyle -> Automatic,
 ItemSize -> Automatic, ItemStyle -> None, Spacings -> Automatic}
```

Thus,

```
p0 = Grid[{{a, b, c}, {d, e}, {f}}, Frame -> All]
```

creates a basic grid. The first row consists of the entries a , b , and c ; the second row d and e ; and the third row f . See Figure 5.4.

You can create quite complex arrays with Grid. For example, elements of grids can be any Mathematica object, including grids.

In the following, we use ExampleData to generate several typical Mathematica objects.

StringTake[string,n]
returns the first n
characters of the
string string.

```
p1 = ExampleData[{"AerialImage", "Earth"}];
p2 = StringTake[ExampleData[{"Text", "GettysburgAddress"}], 100];
p3 = ExampleData[{"Geometry3D", "KleinBottle"}];
p4 = ExampleData[{"Texture", "Bubbles3"}]; .1in
```

a	b	c
d	e	
f		

FIGURE 5.4

A basic grid

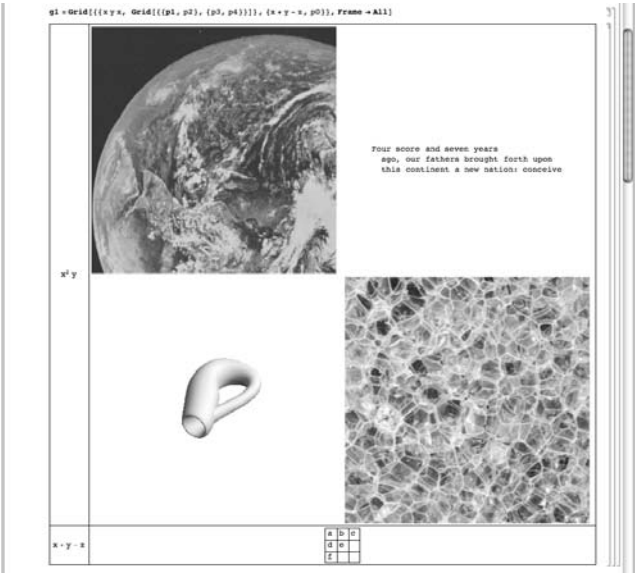


FIGURE 5.5

Very basic grids can appear to be quite complicated

Using our first grid, the previous data, and a few more strings, we create a more sophisticated grid in Figure 5.5.

`g1 = Grid[{{xyx, Grid[{{p1, p2}, {p3, p4}}]}, {x + y - z, p0}}, Frame -> All]`

5.2 LINEAR SYSTEMS OF EQUATIONS

5.2.1 Calculating Solutions of Linear Systems of Equations

To solve the system of linear equations $Ax = b$, where A is the coefficient matrix, b is the known vector, and x is the unknown vector, we often proceed as follows: If A^{-1} exists, then $AA^{-1}x = A^{-1}b$ so $x = A^{-1}b$.

Mathematica offers several commands for solving systems of linear equations, however, that do not depend on the computation of the inverse of **A**. The command

Solve[{eqn1, eqn2, ..., eqnm}, {var1, var2, ..., varn}]

solves an $m \times n$ system of linear equations (m equations and n unknown variables). Note that both the equations and the variables are entered as lists. If one wishes to solve for all variables that appear in a system, the command **Solve[{eqn1, eqn2, ..., eqnn}]** attempts to solve eqn1, eqn2, ..., eqnn for all variables that appear in them. (Remember that a double equals sign (==) must be placed between the left- and right-hand sides of each equation.)

Example 5.2.1 Solve the matrix equation $\begin{pmatrix} 3 & 0 & 2 \\ -3 & 2 & 2 \\ 2 & -3 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ 4 \end{pmatrix}$.

Solution The solution is given by $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 & 0 & 2 \\ -3 & 2 & 2 \\ 2 & -3 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 3 \\ -1 \\ 4 \end{pmatrix}$. We proceed by defining **matrixa** and **b** and then using **Inverse** to calculate **Inverse[matrixa].b**, naming the resulting output {x,y,z}.

matrixa = {{3, 0, 2}, {-3, 2, 2}, {2, -3, 3}};

b = {3, -1, 4};

{x, y, z} = Inverse[matrixa].b

$\left\{ \frac{13}{23}, -\frac{7}{23}, \frac{15}{23} \right\}$

We verify that the result is the desired solution by calculating **matrixa.{x,y,z}**.

Because the result of this procedure is $\begin{pmatrix} 3 \\ -1 \\ 4 \end{pmatrix}$, we conclude that the solution

to the system is $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 13/23 \\ -7/23 \\ 15/23 \end{pmatrix}$.

matrixa.{x, y, z}

{3, -1, 4}

We note that this matrix equation is equivalent to the system of equations

$$3x + 2z = 3$$

$$-3x + 2y + 2z = -1,$$

$$2x - 3y + 3z = 4$$

which we are able to solve with **Solve**. (Note that **Thread[{f1, f2, ...} = {g1, g2, ...}]** returns the system of equations {f1==g1, f2==g2, ...}.)

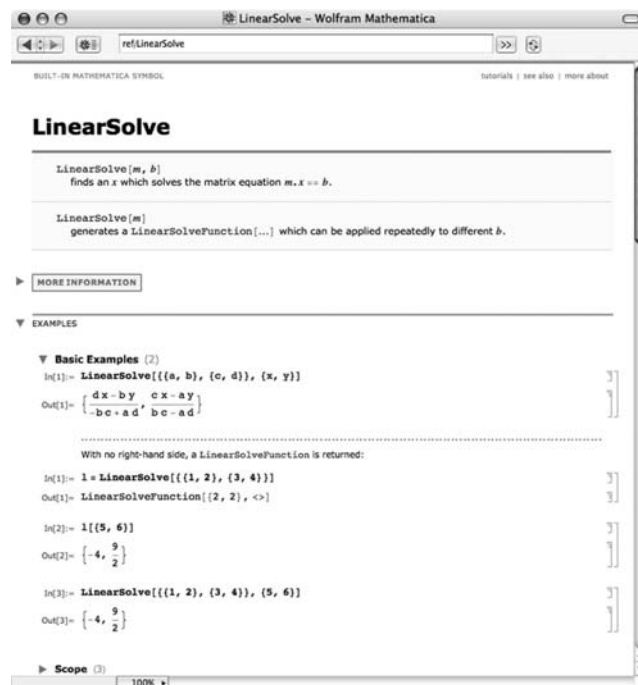
```
Clear[x, y, z]
sys = Thread[matrixa . {x, y, z} == {3, -1, 4}]
{3x + 2z == 3, -3x + 2y + 2z == -1, 2x - 3y + 3z == 4}

Solve[sys]
{{x -> 13/23, z -> 15/23, y -> -7/23}}
```

In addition to using `Solve` to solve a system of linear equations, the command

`LinearSolve[A,b]`

calculates the solution vector x of the system $Ax = b$. `LinearSolve` generally solves a system more quickly than does `Solve`, as we see from the comments in the **Documentation Center**.



Example 5.2.2 Solve the system
$$\begin{cases} x - 2y + z = -4 \\ 3x + 2y - z = 8 \\ -x + 3y + 5z = 0 \end{cases} \quad \text{for } x, y, \text{ and } z.$$

Solution In this case, entering either

Solve[[$x - 2y + z == -4$, $3x + 2y - z == 8$, $-x + 3y + 5z == 0$]]

or

Solve[[$x - 2y + z$, $3x + 2y - z$, $-x + 3y + 5z$]=={-4, 8, 0}]

gives the same result.

Solve[[$x - 2y + z == -4$, $3x + 2y - z == 8$, $-x + 3y + 5z == 0$], { x , y , z }]

{ $\{x \rightarrow 1, y \rightarrow 2, z \rightarrow -1\}$ }

Another way to solve systems of equations is based on the matrix form of the system of equations, $\mathbf{Ax} = \mathbf{b}$. This system of equations is equivalent to the matrix equation

$$\begin{pmatrix} 1 & -2 & 1 \\ 3 & 2 & -1 \\ -1 & 3 & 5 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -4 \\ 8 \\ 0 \end{pmatrix}.$$

The matrix of coefficients in the previous example is entered as **matrixa** along with the vector of right-hand side values **vectorb**. After defining the vector of variables, **vectorx**, the system $\mathbf{Ax} = \mathbf{b}$ is solved explicitly with the command **Solve**.

matrixa = {{1, -2, 1}, {3, 2, -1}, {-1, 3, 5}};

vectorb = {-4, 8, 0};

vectorx = {x1, y1, z1};

Solve[**matrixa**.**vectorx**==**vectorb**, **vectorx**]

{{x1 $\rightarrow$ 1, y1 $\rightarrow$ 2, z1 $\rightarrow$ -1}}

LinearSolve[**matrixa**, **vectorb**]

{1, 2, -1}

Example 5.2.3 Solve the system $\begin{cases} 2x - 4y + z = -1 \\ 3x + y - 2z = 3 \\ -5x + y - 2z = 4 \end{cases}$. Verify that the result returned satisfies the system.

Solution To solve the system using **Solve**, we define **eqs** to be the set of three equations to be solved and **vars** to be the variables x , y , and z and then use **Solve** to solve the set of equations **eqs** for the variables in **vars**. The resulting output is named **sols**.

eqs = { $2x - 4y + z == -1$, $3x + y - 2z == 3$, $-5x + y - 2z == 4$ }; **vars** = { x , y , z };
sols = **Solve**[**eqs**, **vars**]

$$\left\{ \left\{ x \rightarrow -\frac{1}{8}, y \rightarrow -\frac{15}{56}, z \rightarrow -\frac{51}{28} \right\} \right\}$$

To verify that the result given in **sols** is the desired solution, we replace each occurrence of x , y , and z in **eqs** by the values found in **sols** using **ReplaceAll** (/.). Because the result indicates each of the three equations is satisfied, we conclude that the values given in **sols** are the components of the desired solution.

eqs/sols

{ {True, True, True} }

To solve the system using **LinearSolve**, we note that the system is equivalent to the matrix equation $\begin{pmatrix} 2 & -4 & 1 \\ 3 & 1 & -2 \\ -5 & 1 & -2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 4 \end{pmatrix}$, define **matrixa** and **vectorb**, and use **LinearSolve** to solve this matrix equation.

matrixa = {{2, -4, 1}, {3, 1, -2}, {-5, 1, -2}};

vectorb = {-1, 3, 4};

solvevector = LinearSolve[matrixa, vectorb]

$$\left\{ -\frac{1}{8}, -\frac{15}{56}, -\frac{51}{28} \right\}$$

To verify that the results are correct, we compute **matrixa.solvevector**. Because the result is $\begin{pmatrix} -1 \\ 3 \\ 4 \end{pmatrix}$, we conclude that the solution to the system is $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1/8 \\ -15/56 \\ -51/28 \end{pmatrix}$.

matrixa.solvevector

{ -1, 3, 4 }

The command **LinearSolve[A]** returns a function that when given a vector **b** solves the equation **Ax = b**: **LinearSolve[A][b]** returns **x**.

LinearSolve[matrixa]

LinearSolveFunction[{3, 3}, <>]

LinearSolve[matrixa][{-1, 3, 4}]

$$\left\{ -\frac{1}{8}, -\frac{15}{56}, -\frac{51}{28} \right\}$$

Enter indexed variables such $x_1, x_2, \dots, x_n$ as **x[1], x[2], ..., x[n]**. If you need to include the entire list, **Table[x[i], {i, 1, n}]** usually produces the desired result(s).

Example 5.2.4 Solve the system of equations

$$\begin{cases} 4x_1 + 5x_2 - 5x_3 - 8x_4 - 2x_5 = 5 \\ 7x_1 + 2x_2 - 10x_3 - x_4 - 6x_5 = -4 \\ 6x_1 + 2x_2 + 10x_3 - 10x_4 + 7x_5 = -7 \\ -8x_1 - x_2 - 4x_3 + 3x_5 = 5 \\ 8x_1 - 7x_2 - 3x_3 + 10x_4 + 5x_5 = 7 \end{cases}$$

Solution We solve the system in two ways. First, we use `Solve` to solve the system. Note that in this case, we enter the equations in the form

set of left-hand sides == set of right-hand sides

```
Solve[{4x[1] + 5x[2] - 5x[3] - 8x[4] - 2x[5],
      7x[1] + 2x[2] - 10x[3] - x[4] - 6x[5],
      6x[1] + 2x[2] + 10x[3] - 10x[4] + 7x[5],
      -8x[1] - x[2] - 4x[3] + 3x[5],
      8x[1] - 7x[2] - 3x[3] + 10x[4] + 5x[5]} == {5, -4, -7, 5, 7}]
```

```
{ {x[1] -> 1245/6626, x[2] -> 113174/9939, x[3] -> -7457/9939, x[4] -> 38523/6626, x[5] -> 49327/9939 } }
```

We also use `LinearSolve` after defining `matrixa` and `t2`. As expected, in each case, the results are the same.

```
Clear[matrixa]
matrixa = {{4, 5, -5, -8, -2}, {7, 2, -10, -1, -6}, {6, 2, 10, -10, 7},
           {-8, -1, -4, 0, 3}, {8, -7, -3, 10, 5}};
t2 = {5, -4, -7, 5, 7};
LinearSolve[matrixa, t2]

{ 1245/6626, 113174/9939, -7457/9939, 38523/6626, 49327/9939 }
```

5.2.2 Gauss–Jordan Elimination

Given the matrix equation $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix},$$

the $m \times n$ matrix A is called the **coefficient matrix** for the matrix equation $A\mathbf{x} = \mathbf{b}$, and the $m \times (n + 1)$ matrix

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{pmatrix}$$

is called the **augmented** (or **associated**) **matrix** for the matrix equation. We may enter the augmented matrix associated with a linear system of equations directly or we can use commands such as `Join` to help us construct the augmented matrix. For example, if **A** and **B** are rectangular matrices that have the same number of columns, `Join[A,B]` returns $\begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix}$. On the other hand, if **A** and **B** are rectangular matrices that have the same number of rows, `Join[A,B,2]` returns the concatenated matrix $(\mathbf{A} \ \mathbf{B})$.

Example 5.2.5 Solve the system
$$\begin{cases} -2x + y - 2z = 4 \\ 2x - 4y - 2z = -4 \\ x - 4y - 2z = 3 \end{cases}$$
 using Gauss–Jordan elimination.

Solution The system is equivalent to the matrix equation

$$\begin{pmatrix} -2 & 1 & -2 \\ 2 & -4 & -2 \\ 1 & -4 & -2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ -4 \\ 3 \end{pmatrix}.$$

The augmented matrix associated with this system is

$$\begin{pmatrix} -2 & 1 & -2 & 4 \\ 2 & -4 & -2 & -4 \\ 1 & -4 & -2 & 3 \end{pmatrix},$$

which we construct using the command `Join`.

```
matrixa = {{-2, 1, -2}, {2, -4, -2}, {1, -4, -2}};
b = {{4}, {-4}, {3}};
augm = Join[matrixa, b, 2];
MatrixForm[augm]
```

$$\begin{pmatrix} -2 & 1 & -2 & 4 \\ 2 & -4 & -2 & -4 \\ 1 & -4 & -2 & 3 \end{pmatrix}$$

We calculate the solution by row-reducing `augm` using `RowReduce`. Generally, `RowReduce[A]` reduces **A** to **reduced row echelon form**.

```
RowReduce[augm]//MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 & -7 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

From this result, we see that the solution is

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -7 \\ -4 \\ 3 \end{pmatrix}.$$

We verify this by replacing each occurrence of x , y , and z on the left-hand side of the equations by -7 , -4 , and 3 , respectively, and noting that the components of the result are equal to the right-hand side of each equation.

```
Clear[x, y, z]
{-2x + y - 2z, 2x - 4y - 2z, x - 4y - 2z}/.{x -> -7, y -> -4, z -> 3}
{4, -4, 3}
```

In the following example, we carry out the steps of the row reduction process.

Example 5.2.6 Solve

$$-3x + 2y - 2z = -10$$

$$3x - y + 2z = 7$$

$$2x - y + z = 6.$$

Solution The associated matrix is $A = \begin{pmatrix} -3 & 2 & -2 & -10 \\ 3 & -1 & 2 & 7 \\ 2 & -1 & 1 & 6 \end{pmatrix}$, defined in `capa` and then displayed in traditional row-and-column form with `MatrixForm`.

```
Clear[capa]
capa = {{-3, 2, -2, -10}, {3, -1, 2, 7}, {2, -1, 1, 6}};
MatrixForm[capa]
```

$$\begin{pmatrix} -3 & 2 & -2 & -10 \\ 3 & -1 & 2 & 7 \\ 2 & -1 & 1 & 6 \end{pmatrix}$$

We eliminate methodically. First, we multiply row 1 by $-1/3$ so that the first entry in the first column is 1.

```
capa = {-1/3capa[[1]], capa[[2]], capa[[3]]}
{{1, -2/3, 2/3, 10/3}, {3, -1, 2, 7}, {2, -1, 1, 6}}
```

We now eliminate below. First, we multiply row 1 by -3 and add it to row 2 and then we multiply row 1 by -2 and add it to row 3.

```
capa = {capa[[1]], -3capa[[1]] + capa[[2]],
        -2capa[[1]] + capa[[3]]}
{{1, -2/3, 2/3, 10/3}, {0, 1, 0, -3}, {0, 1/3, -1/3, -2/3}}
```

Observe that the first nonzero entry in the second row is 1. We eliminate below this entry by adding $-1/3$ times row 2 to row 3.

```
capa = {capa[[1]], capa[[2]],
        -1/3capa[[2]] + capa[[3]]}
```

$$\left\{ \left\{ 1, -\frac{2}{3}, \frac{2}{3}, \frac{10}{3} \right\}, \{0, 1, 0, -3\}, \{0, 0, -\frac{1}{3}, \frac{1}{3}\} \right\}$$

We multiply the third row by -3 so that the first nonzero entry is 1.

$$\mathbf{capa} = \{\mathbf{capa}[[1]], \mathbf{capa}[[2]], -3\mathbf{capa}[[3]]\}$$

$$\left\{ \left\{ 1, -\frac{2}{3}, \frac{2}{3}, \frac{10}{3} \right\}, \{0, 1, 0, -3\}, \{0, 0, 1, -1\} \right\}$$

This matrix is equivalent to the system

$$\begin{aligned} x - \frac{2}{3}y + \frac{2}{3}z &= \frac{10}{3} \\ y &= -3 \\ z &= -1, \end{aligned}$$

which shows us that the solution is $x = 2$, $y = -3$, $z = -1$.

Working backwards confirms this. Multiplying row 2 by $2/3$ and adding to row 1 and then multiplying row 3 by $-2/3$ and adding to row 1 results in

$$\mathbf{capa} = \{2/3\mathbf{capa}[[2]] + \mathbf{capa}[[1]], \mathbf{capa}[[2]], \mathbf{capa}[[3]]\}$$

$$\mathbf{capa} = \{-2/3\mathbf{capa}[[3]] + \mathbf{capa}[[1]], \mathbf{capa}[[2]], \mathbf{capa}[[3]]\}$$

$$\mathbf{MatrixForm}[\mathbf{capa}]$$

$$\left\{ \left\{ 1, 0, \frac{2}{3}, \frac{4}{3} \right\}, \{0, 1, 0, -3\}, \{0, 0, 1, -1\} \right\}$$

$$\left\{ \left\{ 1, 0, 0, 2 \right\}, \{0, 1, 0, -3\}, \{0, 0, 1, -1\} \right\}$$

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

which is equivalent to the system $x = 2$, $y = -3$, $z = -1$.

Equivalent results are obtained with `RowReduce`.

$$\mathbf{capa} = \{\{-3, 2, -2, -10\}, \{3, -1, 2, 7\}, \{2, -1, 1, 6\}\};$$

$$\mathbf{capa} = \mathbf{RowReduce}[\mathbf{capa}]$$

$$\mathbf{MatrixForm}[\mathbf{capa}]$$

$$\left\{ \left\{ 1, 0, 0, 2 \right\}, \{0, 1, 0, -3\}, \{0, 0, 1, -1\} \right\}$$

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

Finally, we confirm the result directly with `Solve`.

$$\mathbf{Solve}[\{-3x + 2y - 2z == -10, \\ 3x - y + 2z == 7, 2x - y + z == 6\}]$$

$$\{\{x \rightarrow 2, y \rightarrow -3, z \rightarrow -1\}\}$$

Example 5.2.7 Solve

$$-3x_1 + 2x_2 + 5x_3 = -12$$

$$3x_1 - x_2 - 4x_3 = 9$$

$$2x_1 - x_2 - 3x_3 = 7.$$

Solution The associated matrix is $\mathbf{A} = \begin{pmatrix} -3 & 2 & 5 & -12 \\ 3 & -1 & -4 & 9 \\ 2 & -1 & -3 & 7 \end{pmatrix}$, which is reduced to row echelon form with RowReduce.

capa = {{-3, 2, 5, -12}, {3, -1, -4, 9}, {2, -1, -3, 7}};

rrcapa = RowReduce[capa];

MatrixForm[rrcapa]

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

The result shows that the original system is equivalent to

$$\begin{array}{rcl} x_1 - x_3 = 2 & \text{or} & x_1 = 2 + x_3 \\ x_2 + x_3 = -3 & & x_2 = -3 - x_3 \end{array}$$

so x_3 is free. That is, for any real number t , a solution to the system is

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 + t \\ -3 - t \\ t \end{pmatrix} = \begin{pmatrix} 2 \\ -3 \\ 0 \end{pmatrix} + t \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}.$$

The system has infinitely many solutions.

Equivalent results are obtained with Solve.

```
In[15]:= Solve[{-3 x1 + 2 x2 + 5 x3 == -12, 3 x1 - x2 - 4 x3 == 9,
               2 x1 - x2 - 3 x3 == 7}]

Solve::svars : Equations may not give solutions for all "solve" variables. >>

Out[15]:= {{x1 -> 2 + x3, x2 -> -3 - x3}}
```

**Example 5.2.8** Solve

$$-3x_1 + 2x_2 + 5x_3 = -14$$

$$3x_1 - x_2 - 4x_3 = 11$$

$$2x_1 - x_2 - 3x_3 = 8.$$

Solution The associated matrix is $\mathbf{A} = \begin{pmatrix} -3 & 2 & 5 & -14 \\ 3 & -1 & -4 & 11 \\ 2 & -1 & -3 & 8 \end{pmatrix}$, which is reduced to row echelon form with `RowReduce`.

```
Clear[x]
capa = {{-3, 2, 5, -14}, {3, -1, -4, 11}, {2, -1, -3, 8}};
rrcapa = RowReduce[capa];
MatrixForm[rrcapa]
```

$$\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The result shows that the original system is equivalent to

$$x_1 - x_3 = 0$$

$$x_2 + x_3 = 0$$

$$0 = 1.$$

Of course, 0 is not equal to 1: The last equation is false. The system has no solutions.

We check the calculation with `Solve`. In this case, the results indicate that `Solve` cannot find any solutions to the system.

```
Solve[{-3x[1] + 2x[2] + 5x[3] == -14,
3x[1] - x[2] - 4x[3] == 11, 2x[1] - x[2] - 3x[3] == 8}]
{}
```

Generally, if Mathematica returns nothing, the result means either that there is no solution or that Mathematica cannot solve the problem. In such a situation, we must always check using another method.

Example 5.2.9 The **nullspace** of $\mathbf{A}$ is the set of solutions to the system of equations $\mathbf{A}\mathbf{x} = \mathbf{0}$.

Find the nullspace of $\mathbf{A} = \begin{pmatrix} 3 & 2 & 1 & 1 & -2 \\ 3 & 3 & 1 & 2 & -1 \\ 2 & 2 & 1 & 1 & -1 \\ -1 & -1 & 0 & -1 & 0 \\ 5 & 4 & 2 & 2 & -3 \end{pmatrix}$.

Solution Observe that row reducing $(\mathbf{A}|\mathbf{0})$ is equivalent to row reducing $\mathbf{A}$. After defining $\mathbf{A}$, we use `RowReduce` to row reduce $\mathbf{A}$.

```
capa = {{3, 2, 1, 1, -2}, {3, 3, 1, 2, -1},
{2, 2, 1, 1, -1}, {-1, -1, 0, -1, 0},
```

```
{5, 4, 2, 2, -3}];  
RowReduce[capa]//MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The result indicates that the solutions of $\mathbf{Ax} = \mathbf{0}$ are

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} t \\ -s - t \\ s + t \\ s \\ t \end{pmatrix} = s \begin{pmatrix} 0 \\ -1 \\ 1 \\ 1 \\ 0 \end{pmatrix} + t \begin{pmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 1 \end{pmatrix},$$

where s and t are any real numbers. The dimension of the nullspace, the **nullity**, is 2; a basis for the nullspace is

$$\left\{ \begin{pmatrix} 0 \\ -1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

You can use the command **NullSpace[A]** to find a basis of the nullspace of a matrix **A** directly.

```
NullSpace[capa]  
{ {1, -1, 1, 0, 1}, {0, -1, 1, 1, 0} }
```

A is **singular** because $|\mathbf{A}| = 0$.

```
Det[capa]  
0
```

Do *not* use **LinearSolve** on singular matrices, because the results returned may not be (completely) correct.

```
Don't use LinearSolve on non-singular matrices:
```

```
LinearSolve[capa]  
LinearSolve::sing1 :  
The matrix {{3, 2, 1, 1, -2}, {3, 3, 1, 2, -1}, {2, 2, 1, 1, -1}, {-1, -1, 0, -1, 0}, {5, 4, 2, 2, -3}}  
is singular so a factorization will not be saved.  
LinearSolveFunction[{5, 5}, <>]  
  
Det[capa]  
0  
  
LinearSolve[capa, {0, 0, 0, 0, 0}]  
{0, 0, 0, 0, 0}
```

```
LinearSolve[capa, {0, 0, 0, 0, 0}]
{0, 0, 0, 0, 0}
```

5.3 SELECTED TOPICS FROM LINEAR ALGEBRA

5.3.1 Fundamental Subspaces Associated with Matrices

Let $\mathbf{A} = (a_{ij})$ be an $n \times m$ matrix with entry a_{ij} in the i th row and j th column. The **row space** of $\mathbf{A}$, $\text{row}(\mathbf{A})$, is the spanning set of the rows of $\mathbf{A}$; the **column space** of $\mathbf{A}$, $\text{col}(\mathbf{A})$, is the spanning set of the columns of $\mathbf{A}$. If $\mathbf{A}$ is any matrix, then the dimension of the column space of $\mathbf{A}$ is equal to the dimension of the row space of $\mathbf{A}$. The dimension of the row space (column space) of a matrix $\mathbf{A}$ is called the **rank** of $\mathbf{A}$. The **nullspace** of $\mathbf{A}$ is the set of solutions to the system of equations $\mathbf{A}\mathbf{x} = \mathbf{0}$. The nullspace of $\mathbf{A}$ is a subspace and its dimension is called the **nullity** of $\mathbf{A}$. The rank of $\mathbf{A}$ is equal to the number of nonzero rows in the row echelon form of $\mathbf{A}$, and the nullity of $\mathbf{A}$ is equal to the number of zero rows in the row echelon form of $\mathbf{A}$. Thus, if $\mathbf{A}$ is a square matrix, the sum of the rank of $\mathbf{A}$ and the nullity of $\mathbf{A}$ is equal to the number of rows (columns) of $\mathbf{A}$.

1. `NullSpace[A]` returns a list of vectors that form a basis for the nullspace (or kernel) of the matrix $\mathbf{A}$.
2. `RowReduce[A]` yields the reduced row echelon form of the matrix $\mathbf{A}$.

Example 5.3.1 Place the matrix

$$\mathbf{A} = \begin{pmatrix} -1 & -1 & 2 & 0 & -1 \\ -2 & 2 & 0 & 0 & -2 \\ 2 & -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 2 & 2 \\ 1 & -2 & 2 & -2 & 0 \end{pmatrix}$$

in reduced row echelon form. What is the rank of $\mathbf{A}$? Find a basis for the nullspace of $\mathbf{A}$.

Solution We begin by defining the matrix `matrixa`. Then, `RowReduce` is used to place `matrixa` in reduced row echelon form.

```

capa = {{-1, -1, 2, 0, -1}, {-2, 2, 0, 0, -2},
{2, -1, -1, 0, 1}, {-1, -1, 1, 2, 2},
{1, -2, 2, -2, 0}};
RowReduce[capa]//MatrixForm

```

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

Because the row-reduced form of **matrixa** contains four nonzero rows, the rank of **A** is 4 and thus the nullity is 1. We obtain a basis for the nullspace with **NullSpace**.

```

NullSpace[capa]
{{2, 2, 2, 1, 0}}

```

As expected, because the nullity is 1, a basis for the nullspace contains one vector.

Example 5.3.2 Find a basis for the column space of

$$\mathbf{B} = \begin{pmatrix} 1 & -2 & 2 & 1 & -2 \\ 1 & 1 & 2 & -2 & -2 \\ 1 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & -2 & 0 \\ -2 & 1 & 0 & 1 & 2 \end{pmatrix}.$$

Solution A basis for the column space of **B** is the same as a basis for the row space of the transpose of **B**. We begin by defining **matrixb** and then using **Transpose** to compute the transpose of **matrixb**, naming the resulting output **tb**.

```

matrixb = {{1, -2, 2, 1, -2}, {1, 1, 2, -2, -2},
{1, 0, 0, 2, -1}, {0, 0, 0, -2, 0},
{-2, 1, 0, 1, 2}};
tb = Transpose[matrixb]
{{1, 1, 1, 0, -2}, {-2, 1, 0, 0, 1},
{2, 2, 0, 0, 0}, {1, -2, 2, -2, 1}, {-2, -2, -1, 0, 2}}

```

Next, we use **RowReduce** to row reduce **tb** and name the result **rrtb**. A basis for the column space consists of the first four elements of **rrtb**. We also use **Transpose** to show that the first four elements of **rrtb** are the same as the first four columns of the transpose of **rrtb**. Thus, the *j*th column of a matrix **A** can be extracted from **A** with **Transpose[A][[j]]**.

```

rrtb = RowReduce[tb];
Transpose[rrtb]//MatrixForm

```

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

We extract the first four elements of `rrtb` with `Take`. The results correspond to a basis for the column space of **B**.

Take[rrtb, 4]

$\{\{1, 0, 0, 0, -\frac{1}{3}\}, \{0, 1, 0, 0, \frac{1}{3}\}, \{0, 0, 1, 0, -2\}, \{0, 0, 0, 1, -3\}\}$

5.3.2 The Gram–Schmidt Process

A set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ is **orthonormal** means that $\|\mathbf{v}_i\| = 1$ for all values of i and $\mathbf{v}_i \cdot \mathbf{v}_j = 0$ for $i \neq j$. Given a set of linearly independent vectors $S = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$, the set of all linear combinations of the elements of S , $V = \text{span } S$, is a vector space. Note that if S is an orthonormal set and $\mathbf{u} \in \text{span } S$, then $\mathbf{u} = (\mathbf{u} \cdot \mathbf{v}_1) \mathbf{v}_1 + (\mathbf{u} \cdot \mathbf{v}_2) \mathbf{v}_2 + \dots + (\mathbf{u} \cdot \mathbf{v}_n) \mathbf{v}_n$. Thus, we may easily express $\mathbf{u}$ as a linear combination of the vectors in S . Consequently, if we are given any vector space, V , it is frequently convenient to be able to find an orthonormal basis of V . We may use the **Gram–Schmidt process** to find an orthonormal basis of the vector space $V = \text{span } \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$.

We summarize the algorithm of the Gram–Schmidt process so that given a set of n linearly independent vectors $S = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$, where $V = \text{span } \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$, we can construct a set of orthonormal vectors $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ so that $V = \text{span } \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$.

1. Let $\mathbf{u}_1 = \frac{1}{\|\mathbf{v}_1\|} \mathbf{v}_1$;
2. Compute $\text{proj}_{\{\mathbf{u}_1\}} \mathbf{v}_2 = (\mathbf{u}_1 \cdot \mathbf{v}_2) \mathbf{u}_1$, $\mathbf{v}_2 - \text{proj}_{\{\mathbf{u}_1\}} \mathbf{v}_2$, and let

$$\mathbf{u}_2 = \frac{1}{\|\mathbf{v}_2 - \text{proj}_{\{\mathbf{u}_1\}} \mathbf{v}_2\|} (\mathbf{v}_2 - \text{proj}_{\{\mathbf{u}_1\}} \mathbf{v}_2).$$

Then, $\text{span } \{\mathbf{u}_1, \mathbf{u}_2\} = \text{span } \{\mathbf{v}_1, \mathbf{v}_2\}$ and $\text{span } \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{v}_3, \dots, \mathbf{v}_n\} = \text{span } \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$;

3. Generally, for $3 \leq i \leq n$, compute

$$\text{proj}_{\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{i-1}\}} \mathbf{v}_i = (\mathbf{u}_1 \cdot \mathbf{v}_i) \mathbf{u}_1 + (\mathbf{u}_2 \cdot \mathbf{v}_i) \mathbf{u}_2 + \dots + (\mathbf{u}_{i-1} \cdot \mathbf{v}_i) \mathbf{u}_{i-1},$$

$\mathbf{v}_i - \text{proj}_{\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}} \mathbf{v}_i$, and let

$$\mathbf{u}_1 = \frac{1}{\left\| \text{proj}_{\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}} \mathbf{v}_i \right\|} \left(\text{proj}_{\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}} \mathbf{v}_i \right).$$

Then, $\text{span} \{ \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_i \} = \text{span} \{ \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_i \}$ and

$$\text{span} \{ \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_i, \mathbf{v}_{i+1}, \dots, \mathbf{v}_n \} = \text{span} \{ \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_n \};$$

and

4. Because $\text{span} \{ \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n \} = \text{span} \{ \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \}$ and $\{ \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n \}$ is an orthonormal set, $\{ \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n \}$ is an orthonormal basis of V .

The Gram-Schmidt procedure is well-suited to computer arithmetic. The following code performs each step of the Gram-Schmidt process on a set of n linearly independent vectors $\{ \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \}$. At the completion of each step of the procedure, `gramschmidt[vecs]` prints the list of vectors corresponding to $\{ \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_i, \mathbf{v}_{i+1}, \dots, \mathbf{v}_n \}$ and returns the list of vectors $\{ \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n \}$. Note how comments are inserted into the code using `(*...*)`.

```

gramschmidt[vecs_] := Module[{n, proj, u, capw},
(*n represents the number of vectors in
the listvecs*)
n = Length[vecs];
(*proj[v, capw] computes the projection
of v onto capw*)
proj[v_, capw_] :=
  Sum[Length[capw] capw[[i]].v capw[[i]],
    {i, 1, Length[capw]}];
u[1] = vecs[[1]] /
  Sqrt[vecs[[1]].vecs[[1]]];
capw = {};
u[i_] := u[i] = Module[{stepone},
  stepone = vecs[[i]] - proj[vecs[[i]], capw];
  Together [
    Sqrt[stepone.stepone]
  ] ];
Do[
  u[i];
  AppendTo[capw, u[i]];
  Print[Join[capw, Drop[vecs, i]], {i, 1, n - 1}];
  u[n];
  AppendTo[capw, u[n]]

```

Example 5.3.3 Use the Gram–Schmidt process to transform the basis $S = \left\{ \begin{pmatrix} -2 \\ -1 \\ -2 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \\ -2 \end{pmatrix} \right\}$ of $\mathbf{R}^3$ into an orthonormal basis.

Solution We proceed by defining $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ to be the vectors in the basis S and using `gramschmidt[{v1,v2,v3}]` to find an orthonormal basis.

```
v1 = {-2, -1, -2};
v2 = {0, -1, 2};
v3 = {1, 3, -2};
gramschmidt[{v1, v2, v3}]
{{{-2/3, -1/3, -2/3}, {0, -1, 2}, {1, 3, -2}}}
{{{-2/3, -1/3, -2/3}, {-1/3, -2/3, 2/3}, {1, 3, -2}}}
{{{-2/3, -1/3, -2/3}, {-1/3, -2/3, 2/3}, {-2/3, 2/3, 1/3}}}
```

On the first line of output, the result $\{\mathbf{u}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is given; $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{v}_3\}$ appears on the second line; $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ follows on the third.

Example 5.3.4 Compute an orthonormal basis for the subspace of $\mathbf{R}^4$ spanned by the vectors $\begin{pmatrix} 2 \\ 4 \\ 4 \\ 1 \end{pmatrix}$, $\begin{pmatrix} -4 \\ 1 \\ -3 \\ 2 \end{pmatrix}$, and $\begin{pmatrix} 1 \\ 4 \\ 4 \\ -1 \end{pmatrix}$. Also, verify that the basis vectors are orthogonal and have norm 1.

Solution With `gramschmidt`, we compute the orthonormal basis vectors. Note that Mathematica names `oset` the last result returned by `gramschmidt`. The orthogonality of these vectors is then verified. Notice that `Together` is used to simplify the result in the case of `oset[[2]].oset[[3]]`. The norm of each vector is then found to be 1.

```
oset = gramschmidt[{{2, 4, 4, 1}, {-4, 1, -3, 2}, {1, 4, 4, -1}}]
{{ {2/√37, 4/√37, 4/√37, 1/√37}, {-4, 1, -3, 2}, {1, 4, 4, -1} }}
{{ {2/√37, 4/√37, 4/√37, 1/√37},
  {-60√(2/16909), 93/√33818, -55/√33818, 44√(2/16909)}, {1, 4, 4, -1} }}
{{ {2/√37, 4/√37, 4/√37, 1/√37},
  {-60√(2/16909), 93/√33818, -55/√33818, 44√(2/16909)},
  {-449/√934565, 268/√934565, 156/√934565, -798/√934565} }}
```

The three vectors are extracted with `oset` using `oset[[1]]`, `oset[[2]]`, and `oset[[3]]`.

```
oset[[1]].oset[[2]]
oset[[1]].oset[[3]]
oset[[2]].oset[[3]]
0
0
0

Sqrt[oset[[1]].oset[[1]]]
Sqrt[oset[[2]].oset[[2]]]
Sqrt[oset[[3]].oset[[3]]]
1
1
1
```

Mathematica contains functions that perform most of the operations discussed here.

1. `Orthogonalize[{v1,v2,...},Method->GramSchmidt]` returns an orthonormal set of vectors given the set of vectors $\{v_1, v_2, \dots, v_n\}$. Note that this command does not illustrate each step of the Gram-Schmidt procedure as the `gramschmidt` function defined previously.
2. `Normalize[v]` returns $\frac{1}{\|v\|}v$ given the nonzero vector v .
3. `Projection[v1,v2]` returns the projection of v_1 onto v_2 : $\text{proj}_{v_2} v_1 = \frac{v_1 \cdot v_2}{\|v_2\|^2} v_2$.

Thus,

```
Orthogonalize[{{2, 4, 4, 1}, {-4, 1, -3, 2}, {1, 4, 4, -1}},
Method -> "GramSchmidt"]
```

$$\left\{ \left\{ \frac{2}{\sqrt{37}}, \frac{4}{\sqrt{37}}, \frac{4}{\sqrt{37}}, \frac{1}{\sqrt{37}} \right\}, \right. \\ \left\{ -60 \sqrt{\frac{2}{16909}}, \frac{93}{\sqrt{33818}}, -\frac{55}{\sqrt{33818}}, 44 \sqrt{\frac{2}{16909}} \right\}, \\ \left. \left\{ -\frac{449}{\sqrt{934565}}, \frac{268}{\sqrt{934565}}, \frac{156}{\sqrt{934565}}, -\frac{798}{\sqrt{934565}} \right\} \right\}$$

returns an orthonormal basis for the subspace of $\mathbf{R}^4$ spanned by the vectors

$$\begin{pmatrix} 2 \\ 4 \\ 4 \\ 1 \end{pmatrix}, \begin{pmatrix} -4 \\ 1 \\ -3 \\ 2 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 \\ 4 \\ 4 \\ -1 \end{pmatrix}. \text{ The command}$$

Normalize[{2, 4, 4, 1}]

$$\left\{ \frac{2}{\sqrt{37}}, \frac{4}{\sqrt{37}}, \frac{4}{\sqrt{37}}, \frac{1}{\sqrt{37}} \right\}$$

finds a unit vector with the same direction as the vector $\mathbf{v} = \begin{pmatrix} 2 \\ 4 \\ 4 \\ 1 \end{pmatrix}$. Entering

Projection[{2, 4, 4, 1}, {-4, 1, -3, 2}]

$$\left\{ \frac{28}{15}, -\frac{7}{15}, \frac{7}{5}, -\frac{14}{15} \right\}$$

finds the projection of $\mathbf{v} = \begin{pmatrix} 2 \\ 4 \\ 4 \\ 1 \end{pmatrix}$ onto $\mathbf{w} = \begin{pmatrix} -4 \\ 1 \\ -3 \\ 2 \end{pmatrix}$.

5.3.3 Linear Transformations

A function $T: \mathbf{R}^n \rightarrow \mathbf{R}^m$ is a **linear transformation** means that T satisfies the properties $T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$ and $T(c\mathbf{u}) = cT(\mathbf{u})$ for all vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathbf{R}^n$ and all real numbers c . Let $T: \mathbf{R}^n \rightarrow \mathbf{R}^m$ be a linear transformation and suppose $T(\mathbf{e}_1) = \mathbf{v}_1$, $T(\mathbf{e}_2) = \mathbf{v}_2, \dots, T(\mathbf{e}_n) = \mathbf{v}_n$, where $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ represents the standard basis of $\mathbf{R}^n$ and $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ are (column) vectors in $\mathbf{R}^m$. The **associated matrix** of T is the $m \times n$ matrix $\mathbf{A} = (\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_n)$:

$$\text{If } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad T(\mathbf{x}) = T\left(\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}\right) = \mathbf{A}\mathbf{x} = (\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_n) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

Moreover, if $\mathbf{A}$ is any $m \times n$ matrix, then $\mathbf{A}$ is the associated matrix of the linear transformation defined by $T(\mathbf{x}) = \mathbf{A}\mathbf{x}$. In fact, a linear transformation T is completely determined by its action on any basis.

The **kernel** of the linear transformation T , $\ker(T)$, is the set of all vectors $\mathbf{x}$ in $\mathbf{R}^n$ such that $T(\mathbf{x}) = \mathbf{0}$: $\ker(T) = \{\mathbf{x} \in \mathbf{R}^n | T(\mathbf{x}) = \mathbf{0}\}$. The kernel of T is a subspace of $\mathbf{R}^n$. Because $T(\mathbf{x}) = \mathbf{A}\mathbf{x}$ for all $\mathbf{x}$ in $\mathbf{R}^n$, $\ker(T) = \{\mathbf{x} \in \mathbf{R}^n | T(\mathbf{x}) = \mathbf{0}\} = \{\mathbf{x} \in \mathbf{R}^n | \mathbf{A}\mathbf{x} = \mathbf{0}\}$ so the kernel of T is the same as the nullspace of $\mathbf{A}$.

Example 5.3.5 Let $T: \mathbf{R}^5 \rightarrow \mathbf{R}^3$ be the linear transformation defined by $T(\mathbf{x}) = \begin{pmatrix} 0 & -3 & -1 & -3 & -1 \\ -3 & 3 & -3 & -3 & -1 \\ 2 & 2 & -1 & 1 & 2 \end{pmatrix} \mathbf{x}$. (a) Calculate a basis for the kernel of the linear trans-

formation. (b) Determine which of the vectors $\begin{pmatrix} 4 \\ 2 \\ 0 \\ 0 \\ -6 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 2 \\ -1 \\ -2 \\ 3 \end{pmatrix}$ is in the kernel of T .

Solution We begin by defining `matrixA` to be the matrix $\mathbf{A} = \begin{pmatrix} 0 & -3 & -1 & -3 & -1 \\ -3 & 3 & -3 & -3 & -1 \\ 2 & 2 & -1 & 1 & 2 \end{pmatrix}$ and then defining `t`. A basis for the kernel of T is the same as a basis for the nullspace of $\mathbf{A}$ found with `NullSpace`.

```
Clear[t, x, matrixA]
matrixA = {{0, -3, -1, -3, -1}, {-3, 3, -3, -3, -1},
           {2, 2, -1, 1, 2}};
t[x_] = matrixA.x;
NullSpace[matrixA]
{{-2, -1, 0, 0, 3}, {-6, -8, -15, 13, 0}}
```

Because $\begin{pmatrix} 4 \\ 2 \\ 0 \\ 0 \\ -6 \end{pmatrix}$ is a linear combination of the vectors that form a basis for the

kernel, $\begin{pmatrix} 4 \\ 2 \\ 0 \\ 0 \\ -6 \end{pmatrix}$ is in the kernel, whereas $\begin{pmatrix} 1 \\ 2 \\ -1 \\ -2 \\ 3 \end{pmatrix}$ is not. These results are verified by

evaluating `t` for each vector.

```
t[{4, 2, 0, 0, -6}]
{0, 0, 0}

t[{1, 2, -1, -2, 3}]
{-2, 9, 11}
```

Application: Rotations

Let $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ be a vector in $\mathbf{R}^2$ and θ an angle. Then, there are numbers r and ϕ given by $r = \sqrt{x_1^2 + x_2^2}$ and $\phi = \tan^{-1}(x_2/x_1)$ so that $x_1 = r \cos \phi$

and $x_2 = r \sin \phi$. When we rotate $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} r \cos \phi \\ r \sin \phi \end{pmatrix}$ through the angle θ , we obtain the vector $\mathbf{x}' = \begin{pmatrix} r \cos(\theta + \phi) \\ r \sin(\theta + \phi) \end{pmatrix}$. Using the trigonometric identities $\sin(\theta \pm \phi) = \sin \theta \cos \phi \pm \sin \phi \cos \theta$ and $\cos(\theta \pm \phi) = \cos \theta \cos \phi \mp \sin \theta \sin \phi$, we rewrite

$$\begin{aligned} \mathbf{x}' &= \begin{pmatrix} r \cos(\theta + \phi) \\ r \sin(\theta + \phi) \end{pmatrix} = \begin{pmatrix} r \cos \theta \cos \phi - r \sin \theta \sin \phi \\ r \sin \theta \cos \phi + r \cos \theta \sin \phi \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} r \cos \phi \\ r \sin \phi \end{pmatrix} \\ &= \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \end{aligned}$$

Thus, the vector $\mathbf{x}'$ is obtained from $\mathbf{x}$ by computing $\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \mathbf{x}$. Generally, if θ represents an angle, the linear transformation $T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ defined by $T(\mathbf{x}) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \mathbf{x}$ is called the **rotation of $\mathbb{R}^2$ through the angle θ** . We write code to rotate a polygon through an angle θ . The procedure `rotate` uses a list of n points and the rotation matrix defined in `r` to produce a new list of points that are joined using the `Line` graphics directive. Entering

```
Line[{{x1,y1},{x2,y2},..., {xn,yn}}]
```

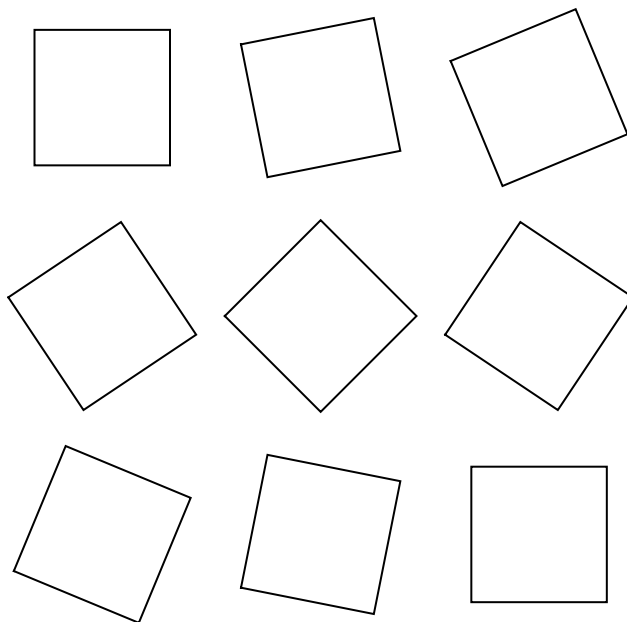
represents the graphics primitive for a line in two dimensions that connects the points listed in $\{\{x_1, y_1\}, \{x_2, y_2\}, \dots, \{x_n, y_n\}\}$. Entering

```
Show[Graphics[Line[{{x1,y1},{x2,y2},..., {xn,yn}}]]]
```

displays the line. This rotation can be determined for one value of θ . However, a more interesting result is obtained by creating a list of rotations for a sequence of angles and then displaying the graphics objects. This is done for $\theta = 0$ to $\theta = \pi/2$ using increments of $\pi/16$. Hence, a list of nine graphs is given for the square with vertices $(-1, 1)$, $(1, 1)$, $(1, -1)$, and $(-1, -1)$ and displayed in Figure 5.6.

$$r[\theta_] = \begin{pmatrix} \text{Cos}[\theta] & -\text{Sin}[\theta] \\ \text{Sin}[\theta] & \text{Cos}[\theta] \end{pmatrix};$$

```
rotate[pts_, angle_] := Module[{newpts},
newpts = Table[r[angle].pts[[i]], {i, 1, Length[pts]}];
newpts = AppendTo[newpts, newpts[[1]];
figure = Line[newpts];
Show[Graphics[figure], AspectRatio -> 1,
PlotRange -> {{-1.5, 1.5}, {-1.5, 1.5}},
DisplayFunction -> Identity]
```

**FIGURE 5.6**

A rotated square

```
graphs = Table[rotate[{{-1, 1}, {1, 1}, {1, -1}, {-1, -1}}, t],
  {t, 0,  $\frac{\pi}{2}$ ,  $\frac{\pi}{16}$ }; array = Partition[graphs, 3];
Show[GraphicsGrid[array]]
```

5.3.4 Eigenvalues and Eigenvectors

Let $\mathbf{A}$ be an $n \times n$ matrix. λ is an **eigenvalue** of $\mathbf{A}$ if there is a *nonzero* vector, $\mathbf{v}$, called an **eigenvector**, satisfying $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. We find the eigenvalues of $\mathbf{A}$ by solving the **characteristic polynomial** $|\mathbf{A} - \lambda\mathbf{I}| = 0$ for λ . Once we find the eigenvalues, the corresponding eigenvectors are found by solving $(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$ for $\mathbf{v}$.

If $\mathbf{A}$ is $n \times n$, `Eigenvalues[A]` finds the eigenvalues of $\mathbf{A}$, `Eigenvectors[A]` finds the eigenvectors, and `Eigensystem[A]` finds the eigenvalues and corresponding eigenvectors. `CharacteristicPolynomial[A, lambda]` finds the characteristic polynomial of $\mathbf{A}$ as a function of λ .

Example 5.3.6 Find the eigenvalues and corresponding eigenvectors for each of the following

matrices: (a) $\mathbf{A} = \begin{pmatrix} -3 & 2 \\ 2 & -3 \end{pmatrix}$, (b) $\mathbf{A} = \begin{pmatrix} 1 & -1 \\ 1 & 3 \end{pmatrix}$, (c) $\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$, and (d) $\mathbf{A} = \begin{pmatrix} -1/4 & 2 \\ -8 & -1/4 \end{pmatrix}$.

Solution (a) We begin by finding the eigenvalues. Solving

$$|\mathbf{A} - \lambda \mathbf{I}| = \begin{vmatrix} -3 - \lambda & 2 \\ 2 & -3 - \lambda \end{vmatrix} = \lambda^2 + 6\lambda + 5 = 0$$

gives us $\lambda_1 = -5$ and $\lambda_2 = -1$.

Observe that the same results are obtained using **CharacteristicPolynomial** and **Eigenvalues**.

```
capa = {{-3, 2}, {2, -3}};
CharacteristicPolynomial[capa, λ]
5 + 6λ + λ2

e1 = Eigenvalues[capa]
{-5, -1}
```

We now find the corresponding eigenvectors. Let $\mathbf{v}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ be an eigenvector corresponding to λ_1 ; then

$$\begin{aligned} (\mathbf{A} - \lambda_1 \mathbf{I}) \mathbf{v}_1 &= \mathbf{0} \\ \left[\begin{pmatrix} -3 & 2 \\ 2 & -3 \end{pmatrix} - (-5) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right] \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned}$$

which row reduces to

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

That is, $x_1 + y_1 = 0$ or $x_1 = -y_1$. Hence, for any value of $y_1 \neq 0$,

$$\mathbf{v}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} -y_1 \\ y_1 \end{pmatrix} = y_1 \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

is an eigenvector corresponding to λ_1 . Of course, this represents infinitely many vectors. However, they are all linearly dependent. Choosing $y_1 = 1$ yields $\mathbf{v}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$. Note that you might have chosen $y_1 = -1$ and obtained $\mathbf{v}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$. However, both of our results are “correct” because these vectors are linearly dependent.

Similarly, letting $\mathbf{v}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ be an eigenvector corresponding to λ_2 , we solve $(\mathbf{A} - \lambda_2 \mathbf{I}) \mathbf{v}_2 = \mathbf{0}$:

$$\begin{pmatrix} -2 & 2 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Thus, $x_2 - y_2 = 0$ or $x_2 = y_2$. Hence, for any value of $y_2 \neq 0$,

$$\mathbf{v}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} y_2 \\ y_2 \end{pmatrix} = y_2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

is an eigenvector corresponding to λ_2 . Choosing $y_2 = 1$ yields $\mathbf{v}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. We confirm these results using `RowReduce`.

```
i2 = IdentityMatrix[2];
ev1 = capa - e1[[1]] i2
{{2, 2}, {2, 2}}

RowReduce[ev1]
{{1, 1}, {0, 0}}

ev2 = capa - e1[[2]] i2;
RowReduce[ev2]
{{1, -1}, {0, 0}}
```

We obtain the same results using `Eigenvectors` and `Eigensystem`.

```
Eigenvectors[capa]
Eigensystem[capa]
{{-1, 1}, {1, 1}}
{{-5, -1}, {{-1, 1}, {1, 1}}}
```

(b) In this case, we see that $\lambda = 2$ has multiplicity 2. There is only one linearly independent eigenvector, $\mathbf{v} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$, corresponding to λ .

```
capa = {{1, -1}, {1, 3}};
Factor[CharacteristicPolynomial[capa, λ]]
Eigenvectors[capa]
Eigensystem[capa]
(-2 + λ)2
{{-1, 1}, {0, 0}}
{{2, 2}, {{-1, 1}, {0, 0}}}
```

(c) The eigenvalue $\lambda_1 = 2$ has corresponding eigenvector $\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. The eigenvalue $\lambda_{2,3} = -1$ has multiplicity 2. In this case, there are two linearly independent eigenvectors corresponding to this eigenvalue: $\mathbf{v}_2 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$ and $\mathbf{v}_3 = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}$.

```
capa = {{0, 1, 1}, {1, 0, 1}, {1, 1, 0}};
Factor[CharacteristicPolynomial[capa, λ]]
```

Eigenvectors[capa]**Eigensystem[capa]**

$$-(-2 + \lambda)(1 + \lambda)^2$$

$$\{\{1, 1, 1\}, \{-1, 0, 1\}, \{-1, 1, 0\}\}$$

$$\{\{2, -1, -1\}, \{\{1, 1, 1\}, \{-1, 0, 1\}, \{-1, 1, 0\}\}\}$$

(d) In this case, the eigenvalues $\lambda_{1,2} = -\frac{1}{4} \pm 4i$ are complex conjugates. We see that the eigenvectors $\mathbf{v}_{1,2} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \pm \begin{pmatrix} 1 \\ 0 \end{pmatrix} i$ are complex conjugates as well.

capa = {{-1/4, 2}, {-8, -1/4}};**Factor[CharacteristicPolynomial[capa, λ],****GaussianIntegers → True]****Eigenvectors[capa]****Eigensystem[capa]**

$$\frac{1}{16}((1 - 16i) + 4\lambda)((1 + 16i) + 4\lambda)$$

$$\{\{-\frac{i}{2}, 1\}, \{\frac{i}{2}, 1\}\}$$

$$\{\{-\frac{1}{4} + 4i, -\frac{1}{4} - 4i\}, \{\{-\frac{i}{2}, 1\}, \{\frac{i}{2}, 1\}\}\}$$

5.3.5 Jordan Canonical Form

Let $\mathbf{N}_k = (n_{ij}) = \begin{cases} 1, j = i + 1 \\ 0, \text{otherwise} \end{cases}$ represent a $k \times k$ matrix with the indicated elements. The $k \times k$ **Jordan block matrix** is given by $\mathbf{B}(\lambda) = \lambda \mathbf{I} + \mathbf{N}_k$, where λ is a constant:

$$\mathbf{N}_k = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{B}(\lambda) = \lambda \mathbf{I} + \mathbf{N}_k = \begin{pmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & \lambda \end{pmatrix}.$$

Hence, $\mathbf{B}(\lambda)$ can be defined as $\mathbf{B}(\lambda) = (b_{ij}) = \begin{cases} \lambda, i = j \\ 1, j = i + 1 \\ 0, \text{otherwise} \end{cases}$. A **Jordan matrix** has the form

$$\mathbf{J} = \begin{pmatrix} \mathbf{B}_1(\lambda) & 0 & \cdots & 0 \\ 0 & \mathbf{B}_2(\lambda) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{B}_n(\lambda) \end{pmatrix},$$

where the entries $\mathbf{B}_j(\lambda)$, $j = 1, 2, \dots, n$ represent Jordan block matrices.

Suppose that A is an $n \times n$ matrix. Then there is an invertible $n \times n$ matrix C such that $C^{-1}AC = J$, where J is a Jordan matrix with the eigenvalues of A as diagonal elements. The matrix J is called the **Jordan canonical form** of A . The command

JordanDecomposition[m]

yields a list of matrices $\{s, j\}$ such that $m = s.j.Inverse[s]$ and j is the Jordan canonical form of the matrix m .

For a given matrix A , the unique monic polynomial q of least degree satisfying $q(A) = 0$ is called the **minimal polynomial** of A . Let p denote the characteristic polynomial of A . Because $p(A) = 0$, it follows that q divides p . We can use the Jordan canonical form of a matrix to determine its minimal polynomial.

Example 5.3.7 Find the Jordan canonical form, J_A , of $A = \begin{pmatrix} 2 & 9 & -9 \\ 0 & 8 & -6 \\ 0 & 9 & -7 \end{pmatrix}$.

Solution After defining `matrixa`, we use `JordanDecomposition` to find the Jordan canonical form of `a` and name the resulting output `ja`.

```
matrixa = {{2, 9, -9}, {0, 8, -6}, {0, 9, -7}};
ja = JordanDecomposition[matrixa]
{{{3, 0, 1}, {2, 1, 0}, {3, 1, 0}},
 {{-1, 0, 0}, {0, 2, 0}, {0, 0, 2}}}
```

The Jordan matrix corresponds to the second element of `ja` extracted with `ja[[2]]` and displayed in `MatrixForm`.

`ja[[2]]//MatrixForm`

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

We also verify that the matrices `ja[[1]]` and `ja[[2]]` satisfy

`matrixa = ja[[1]].ja[[2]].Inverse[ja[[1]]].`

```
ja[[1]].ja[[2]].Inverse[ja[[1]]]
{{2, 9, -9}, {0, 8, -6}, {0, 9, -7}}
```

Next, we use `CharacteristicPolynomial` to find the characteristic polynomial of `matrixa` and then verify that `matrixa` satisfies its characteristic polynomial.

```
p = CharacteristicPolynomial[matrixa, x]
-4 + 3x^2 - x^3
```

```

-4IdentityMatrix[3] + 3MatrixPower[matrixa, 2] -
  MatrixPower[matrixa, 3]
{{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}

```

From the Jordan form, we see that the minimal polynomial of $\mathbf{A}$ is $(x+1)(x-2)$. We define the minimal polynomial to be q and then verify that matrixa satisfies its minimal polynomial.

```

q = Expand[(x + 1)(x - 2)]
-2 - x + x^2

```

```

-2 IdentityMatrix[3] - matrixa + MatrixPower[matrixa, 2]
{{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}

```

As expected, q divides p .

```

Cancel[p/q]
2 - x

```

Example 5.3.8 If $\mathbf{A} = \begin{pmatrix} 3 & 8 & 6 & -1 \\ -3 & 2 & 0 & 3 \\ 3 & -3 & -1 & -3 \\ 4 & 8 & 6 & -2 \end{pmatrix}$, find the characteristic and minimal polynomials of $\mathbf{A}$.

Solution As in the previous example, we first define matrixa and then use `JordanDecomposition` to find the Jordan canonical form of $\mathbf{A}$.

```

matrixa = {{3, 8, 6, -1}, {-3, 2, 0, 3}, {3, -3, -1, -3},
  {4, 8, 6, -2}}; ja = JordanDecomposition[matrixa]
{{{3, -1, 1, 0}, {-1, -1, 0, 1/2},
  {0, 2, 0, -1/2}, {4, 0, 1, 0}},
  {{-1, 0, 0, 0}, {0, -1, 0, 0}, {0, 0, 2, 1}, {0, 0, 0, 2}}}

```

The Jordan canonical form of $\mathbf{A}$ is the second element of ja , extracted with `ja[[2]]` and displayed in `MatrixForm`.

```

ja[[2]]//MatrixForm

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


```

From this result, we see that the minimal polynomial of $\mathbf{A}$ is $(x+1)(x-2)^2$. We define q to be the minimal polynomial of $\mathbf{A}$ and then verify that matrixa satisfies q .

```
q = Expand[(x - 2)^2(x + 1)]
4 - 3x^2 + x^3
```

```
4 IdentityMatrix[4] - 3 MatrixPower[matrixa, 2] +
  MatrixPower[matrixa, 3]
{{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}}
```

The characteristic polynomial is obtained next and named **p**. As expected, **q** divides **p**, verified with **Cancel**.

```
p = CharacteristicPolynomial[matrixa, x]
4 + 4x - 3x^2 - 2x^3 + x^4

Cancel[p/q]
1 + x
```

5.3.6 The QR Method

The **conjugate transpose** (or **Hermitian adjoint matrix**) of the $m \times n$ complex matrix **A** which is denoted by $\mathbf{A}^*$ is the transpose of the complex conjugate of **A**. Symbolically, we have $\mathbf{A}^* = (\mathbf{A})^t$. A complex matrix **A** is **unitary** if $\mathbf{A}^* = \mathbf{A}^{-1}$. Given a matrix **A**, there is a unitary matrix **Q** and an upper triangular matrix **R** such that $\mathbf{A} = \mathbf{QR}$. The product matrix **QR** is called the **QR factorization of A**. The command

QRDecomposition[N[m]]

determines the QR decomposition of the matrix **m** by returning the list **{q,r}**, where **q** is an orthogonal matrix, **r** is an upper triangular matrix, and $\mathbf{m} = \text{Transpose}[\mathbf{q}].\mathbf{r}$.

Example 5.3.9 Find the QR factorization of the matrix $\mathbf{A} = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix}$.

Solution We define **matrixa** and then use **QRDecomposition** to find the QR decomposition of **matrixa**, naming the resulting output **qrm**.

```
matrixa = {{4, -1, 1}, {-1, 4, 1}, {1, 1, 4}};
qrm = QRDecomposition[N[matrixa]]
{{{ -0.942809, 0.235702, -0.235702},
 { -0.142134, -0.92387, -0.355335},
 { -0.301511, -0.301511, 0.904534}},
 {{ -4.24264, 1.64992, -1.64992},
 {0., -3.90868, -2.48734}, {0., 0., 3.01511}}}
```

The first matrix in **qrm** is extracted with **qrm[[1]]** and the second with **qrm[[2]]**.

qrm[[1]]//MatrixForm

$$\begin{pmatrix} -0.942809 & 0.235702 & -0.235702 \\ -0.142134 & -0.92387 & -0.355335 \\ -0.301511 & -0.301511 & 0.904534 \end{pmatrix}$$

qrm[[2]]//MatrixForm

$$\begin{pmatrix} -4.24264 & 1.64992 & -1.64992 \\ 0. & -3.90868 & -2.48734 \\ 0. & 0. & 3.01511 \end{pmatrix}$$

We verify that the results returned are the QR decomposition of $\mathbf{A}$.

Transpose[qrm[[1]]].qrm[[2]]//MatrixForm

$$\begin{pmatrix} 4. & -1. & 1. \\ -1. & 4. & 1. \\ 1. & 1. & 4. \end{pmatrix}$$

One of the most efficient and most widely used methods for numerically calculating the eigenvalues of a matrix is the QR method. Given a matrix $\mathbf{A}$, there is a Hermitian matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$ such that $\mathbf{A} = \mathbf{QR}$. If we define a sequence of matrices $\mathbf{A}_1 = \mathbf{A}$, factored as $\mathbf{A}_1 = \mathbf{Q}_1\mathbf{R}_1$; $\mathbf{A}_2 = \mathbf{R}_1\mathbf{Q}_1$, factored as $\mathbf{A}_2 = \mathbf{R}_2\mathbf{Q}_2$; $\mathbf{A}_3 = \mathbf{R}_2\mathbf{Q}_2$, factored as $\mathbf{A}_3 = \mathbf{R}_3\mathbf{Q}_3$; and in general, $\mathbf{A}_k = \mathbf{R}_{k+1}\mathbf{Q}_{k+1}$, $k = 1, 2, \dots$, then the sequence $\{\mathbf{A}_n\}$ converges to a triangular matrix with the eigenvalues of $\mathbf{A}$ along the diagonal or to a nearly triangular matrix from which the eigenvalues of $\mathbf{A}$ can be calculated rather easily.

Example 5.3.10 Consider the 3×3 matrix $\mathbf{A} = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix}$. Approximate the eigenvalues of $\mathbf{A}$ with the QR method.

Solution We define the sequence $\mathbf{a}$ and $\mathbf{qr}$ recursively. We define $\mathbf{a}$ using the form $\mathbf{a}[\mathbf{n_}] := \mathbf{a}[\mathbf{n}] = \dots$ and $\mathbf{qr}$ using the form $\mathbf{qr}[\mathbf{n_}] := \mathbf{qr}[\mathbf{n}] = \dots$ so that Mathematica “remembers” the values of $\mathbf{a}$ and $\mathbf{qr}$ computed, and thus Mathematica avoids recomputing values previously computed. This is of particular advantage when computing $\mathbf{a}[\mathbf{n}]$ and $\mathbf{qr}[\mathbf{n}]$ for large values of n .

```
matrixa = {{4, -1, 1}, {-1, 4, 1}, {1, 1, 4}};
a[1] = N[matrixa];
qr[1] = QRDecomposition[a[1]];
```

```
a[n_] := a[n] = qr[n - 1][[2]].Transpose[qr[n - 1][[1]]];
qr[n_] := qr[n] = QRDecomposition[a[n]];
```

We illustrate $\mathbf{a}[n]$ and $\mathbf{qr}[n]$ by computing $\mathbf{qr}[9]$ and $\mathbf{a}[10]$. Note that computing $\mathbf{a}[10]$ requires the computation of $\mathbf{qr}[9]$. From the results, we suspect that the eigenvalues of $\mathbf{A}$ are 5 and 2.

```
qr[9];
a[10]//MatrixForm
```

$$\begin{pmatrix} 5. & -1.7853841942690367^{**\wedge} - 7 & -0.000556091 \\ -1.7853841949997794^{**\wedge} - 7 & 5. & -0.000963178 \\ -0.000556091 & -0.000963178 & 2. \end{pmatrix}$$

Next, we compute $\mathbf{a}[n]$ for $n = 5, 10$, and 15 , displaying the result in **TableForm**. We obtain further evidence that the eigenvalues of $\mathbf{A}$ are 5 and 2.

```
Table[a[n]//MatrixForm, {n, 5, 15, 5}]/TableForm
```

$$\begin{array}{ccc} 4.99902 & -0.001701 & 0.0542614 \\ -0.001701 & 4.99706 & 0.0939219 \\ 0.0542614 & 0.0939219 & 2.00393 \end{array}$$

$$\begin{array}{ccc} 5. & -1.7853841942690367^{**\wedge} - 7 & -0.000556091 \\ -1.7853841949997794^{**\wedge} - 7 & 5. & -0.000963178 \\ -0.000556091 & -0.000963178 & 2. \end{array}$$

$$\begin{array}{ccc} 5. & -1.872117829091164^{**\wedge} - 11 & 5.694375936943897^{**\wedge} - 6 \\ -1.8721251221839952^{**\wedge} - 11 & 5. & 9.862948440894718^{**\wedge} - 6 \\ 5.69437593740387^{**\wedge} - 6 & 9.862948440910032^{**\wedge} - 6 & 2. \end{array}$$

We verify that the eigenvalues of $\mathbf{A}$ are indeed 5 and 2 with **Eigenvalues**.

```
Eigenvalues[matrixa]
{5, 5, 2}
```

5.4 MAXIMA AND MINIMA USING LINEAR PROGRAMMING

5.4.1 The Standard Form of a Linear Programming Problem

We call the linear programming problem of the following form the **standard form** of the linear programming problem:

$$\begin{array}{l} \text{Minimize } Z = \underbrace{c_1x_1 + c_2x_2 + \cdots + c_nx_n}_{\text{function}}, \text{ subject to the restrictions} \\ \left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \end{array} \right. \end{array} \quad (5.2)$$

and $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$.

The command

Minimize[{function, inequalities}, {variables}]

solves the standard form of the linear programming problem. Similarly, the command

Maximize[{function, inequalities}, {variables}]

solves the linear programming problem: Maximize $Z = \underbrace{c_1x_1 + c_2x_2 + \cdots + c_nx_n}_{\text{function}}$, subject to the restrictions

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \end{cases}$$

and $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$.

Example 5.4.1 Maximize $Z(x_1, x_2, x_3) = 4x_1 - 3x_2 + 2x_3$ subject to the constraints $3x_1 - 5x_2 + 2x_3 \leq 60$, $x_1 - x_2 + 2x_3 \leq 10$, $x_1 + x_2 - x_3 \leq 20$, and x_1, x_2, x_3 all nonnegative.

Solution In order to solve a linear programming problem with Mathematica, the variables $\{x_1, x_2, x_3\}$ and objective function $z[x_1, x_2, x_3]$ are first defined. In an effort to limit the amount of typing required to complete the problem, the set of inequalities is assigned the name `ineqs` while the set of variables is called `vars`. The symbol “ $\leq$ ”, obtained by typing the “ $<$ ” key and then the “ $=$ ” key, represents “less than or equal to” and is used in `ineqs`. Hence, the maximization problem is solved with the command

Maximize[{z[x1, x2, x3], ineqs}, vars].

Clear[x1, x2, x3, z, ineqs, vars]

vars = {x1, x2, x3};

z[x1_, x2_, x3_] = 4x1 - 3x2 + 2x3;

ineqs = {3x1 - 5x2 + x3 ≤ 60, x1 - x2 + 2x3 ≤ 10, x1 + x2 - x3 ≤ 20,

x1 ≥ 0, x2 ≥ 0, x3 ≥ 0};

Maximize[{z[x1, x2, x3], ineqs}, vars]

{45, {x1 → 15, x2 → 5, x3 → 0}}

The solution gives the maximum value of z subject to the given constraints as well as the values of x_1 , x_2 , and x_3 that maximize z . Thus, we see that the maximum value of Z is 45 if $x_1 = 15$, $x_2 = 5$, and $x_3 = 0$.

We demonstrate the use of `Minimize` in the following example.

Example 5.4.2 Minimize $Z(x, y, z) = 4x - 3y + 2z$ subject to the constraints $3x - 5y + z \leq 60$, $x - y + 2z \leq 10$, $x + y - z \leq 20$, and x, y, z all nonnegative.

Solution After clearing all previously used names of functions and variable values, the variables, objective function, and set of constraints for this problem are defined and entered as they were in the first example. By using `Minimize`, the minimum value of the objective function is obtained as well as the variable values that give this minimum.

```
Clear[x1, x2, x3, z, ineqs, vars]
vars = {x1, x2, x3};
z[x1_, x2_, x3_] = 4x1 - 3x2 + 2x3;
ineqs = {3x1 - 5x2 + x3 ≤ 60, x1 - x2 + 2x3 ≤ 10, x1 + x2 - x3 ≤ 20,
        x1 ≥ 0, x2 ≥ 0, x3 ≥ 0};
Minimize[{z[x1, x2, x3], ineqs}, vars]
{-90, {x1 → 0, x2 → 50, x3 → 30}}
```

We conclude that the minimum value is -90 and occurs if $x_1 = 0$, $x_2 = 50$, and $x_3 = 30$.

5.4.2 The Dual Problem

Given the standard form of the linear programming problem in equations (5.4.1), the **dual problem** is as follows: “Maximize $Y = \sum_{i=1}^m b_i y_i$ subject to the constraints $\sum_{i=1}^m a_{ij} y_i \leq c_j$ for $j = 1, 2, \dots, n$ and $y_i \geq 0$ for $i = 1, 2, \dots, m$.” Similarly, for the problem, “Maximize $Z = \sum_{j=1}^n c_j x_j$ subject to the constraints $\sum_{j=1}^n a_{ij} x_j \leq b_i$ for $i = 1, 2, \dots, m$ and $x_j \geq 0$ for $j = 1, 2, \dots, n$,” the dual problem is as follows: “Minimize $Y = \sum_{i=1}^m b_i y_i$ subject to the constraints $\sum_{i=1}^m a_{ij} y_i \geq c_j$ for $j = 1, 2, \dots, n$ and $y_i \geq 0$ for $i = 1, 2, \dots, m$.”

Example 5.4.3 Maximize $Z = 6x + 8y$ subject to the constraints $5x + 2y \leq 20$, $x + 2y \leq 10$, $x \geq 0$, and $y \geq 0$. State the dual problem and find its solution.

Solution First, the original (or *primal*) problem is solved. The objective function for this problem is represented by `zx`. Finally, the set of inequalities for the primal is defined to be `ineqsx`. Using the command

```
Maximize[{zx, ineqsx}, {x[1], x[2]}]
```

the maximum value of `zx` is found to be 45.

```
Clear[zx, zy, x, y, valsx, valsy, ineqsx, ineqsy]
zx = 6x[1] + 8x[2];
```

ineqsx = { $5x[1] + 2x[2] \leq 20$, $x[1] + 2x[2] \leq 10$, $x[1] \geq 0$,
 $x[2] \geq 0$ };

Maximize[{**zx**, **ineqsx**}, {**x**[1], **x**[2]}]

{45, { $x[1] \rightarrow \frac{5}{2}$, $x[2] \rightarrow \frac{15}{4}$ }}}

Because in this problem we have $c_1 = 6$, $c_2 = 8$, $b_1 = 20$, and $b_2 = 10$, the dual problem is as follows: Minimize $Z = 20y_1 + 10y_2$ subject to the constraints $5y_1 + y_2 \geq 6$, $2y_1 + 2y_2 \geq 8$, $y_1 \geq 0$, and $y_2 \geq 0$. The dual is solved in a similar manner by defining the objective function **zy** and the collection of inequalities **ineqsy**. The minimum value obtained by **zy** subject to the constraints **ineqsy** is 45, which agrees with the result of the primal and is found with

Minimize[{**zy**, **ineqsy**}, {**y**[1], **y**[2]}]

zy = $20y[1] + 10y[2]$;

ineqsy = { $5y[1] + y[2] \geq 6$, $2y[1] + 2y[2] \geq 8$,

$y[1] \geq 0$, $y[2] \geq 0$ };

Minimize[{**zy**, **ineqsy**}, {**y**[1], **y**[2]}]

{45, { $y[1] \rightarrow \frac{1}{2}$, $y[2] \rightarrow \frac{7}{2}$ }}}

Of course, linear programming models can involve numerous variables. Consider the following: Given the standard form linear programming prob-

lem in equations (5.4.1), let $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$, $\mathbf{c} = (c_1 \ c_2 \ \cdots \ c_n)$,

and $\mathbf{A}$ denote the $m \times n$ matrix $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$. Then the stan-

dard form of the linear programming problem is equivalent to finding the vector $\mathbf{x}$ that maximizes $Z = \mathbf{c} \cdot \mathbf{x}$ subject to the restrictions $\mathbf{Ax} \geq \mathbf{b}$ and $x_1 \geq 0$, $x_2 \geq 0$, ..., $x_n \geq 0$. The dual problem is "Minimize $Y = \mathbf{y} \cdot \mathbf{b}$ where $\mathbf{y} = (y_1 \ y_2 \ \cdots \ y_m)$ subject to the restrictions $\mathbf{yA} \leq \mathbf{c}$ (componentwise) and $y_1 \geq 0$, $y_2 \geq 0$, ..., $y_m \geq 0$."

The command

LinearProgramming[**c**, **A**, **b**]

finds the vector $\mathbf{x}$ that minimizes the quantity $Z = \mathbf{c} \cdot \mathbf{x}$ subject to the restrictions $\mathbf{Ax} \geq \mathbf{b}$ and $\mathbf{x} \geq 0$. **LinearProgramming** does not yield the minimum value of Z as did **Minimize** and **Maximize**, and the value must be determined from the resulting vector.

Example 5.4.4 Maximize $Z = 5x_1 - 7x_2 + 7x_3 + 5x_4 + 6x_5$ subject to the constraints $2x_1 + 3x_2 + 3x_3 + 2x_4 + 2x_5 \geq 10$, $6x_1 + 5x_2 + 4x_3 + x_4 + 4x_5 \geq 30$, $-3x_1 - 2x_2 - 3x_3 - 4x_4 \geq -5$, $-x_1 - x_2 - x_4 \geq -10$, and $x_i \geq 0$ for $i = 1, 2, 3, 4$, and 5 . State the dual problem. What is its solution?

Solution For this problem, $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} 10 \\ 30 \\ -5 \\ -10 \end{pmatrix}$, $\mathbf{c} = (5 \ -7 \ 7 \ 5 \ 6)$, and

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 3 & 2 & 2 \\ 6 & 5 & 4 & 1 & 4 \\ -3 & -2 & -3 & -4 & 0 \\ -1 & -1 & 0 & -1 & 0 \end{pmatrix}. \text{ First, the vectors } \mathbf{c} \text{ and } \mathbf{b} \text{ are entered and then matrix}$$

$\mathbf{A}$ is entered and named `matrixa`.

```
Clear[matrixa, x, y, c, b]
c = {5, -7, 7, 5, 6}; b = {10, 30, -5, -10};
matrixa = {{2, 3, 3, 2, 2}, {6, 5, 4, 1, 4},
           {-3, -2, -3, -4, 0}, {-1, -1, 0, -1, 0}};
```

Next, we use `Array[x, 5]` to create the list of five elements $\{x[1], x[2], \dots, x[5]\}$ named `xvec`. The command `Table[x[i], {i, 1, 5}]` returns the same list. These variables must be defined before attempting to solve this linear programming problem.

```
xvec = Array[x, 5]
{x[1], x[2], x[3], x[4], x[5]}
```

After entering the objective function coefficients with the vector $\mathbf{c}$, the matrix of coefficients from the inequalities with `matrixa`, and the right-hand side values found in $\mathbf{b}$, the problem is solved with

LinearProgramming[c, matrixa, b]

The solution is called `xvec`. Hence, the maximum value of the objective function is obtained by evaluating the objective function at the variable values that yield a maximum. Because these values are found in `xvec`, the maximum is determined with the dot product of the vector $\mathbf{c}$ and the vector `xvec`. (Recall that this product is entered as `c.xvec`.) This value is found to be $35/4$.

```
xvec = LinearProgramming[c, matrixa, b]
{0, 5/2, 0, 0, 35/8}
```

```
c.xvec
35/4
```

Because the dual of the problem is “Minimize the number $Y = \mathbf{y} \cdot \mathbf{b}$ subject to the restrictions $\mathbf{y} \cdot \mathbf{A} < \mathbf{c}$ and $\mathbf{y} > 0$,” we use Mathematica to calculate $\mathbf{y} \cdot \mathbf{b}$ and $\mathbf{y} \cdot \mathbf{A}$. A list of the dual variables $\{y[1], y[2], y[3], y[4]\}$ is created with `Array[y, 4]`. This list includes

four elements because there are four constraints in the original problem. The objective function of the dual problem is therefore found with **yvec.b**, and the left-hand sides of the set of inequalities are given with **yvec.matrixa**.

```
yvec = Array[y, 4]
{y[1], y[2], y[3], y[4]}
yvec.b
10y[1] + 30y[2] - 5y[3] - 10y[4]
yvec.matrixa
{2y[1] + 6y[2] - 3y[3] - y[4],
3y[1] + 5y[2] - 2y[3] - y[4],
3y[1] + 4y[2] - 3y[3],
2y[1] + y[2] - 4y[3] - y[4],
2y[1] + 4y[2]}
```

Hence, we may state the dual problem as follows:

Minimize $Y = 10y_1 + 30y_2 - 5y_3 - 10y_4$ subject to the constraints

$$\begin{cases} 2y_1 + 6y_2 - 3y_3 - y_4 \leq 5 \\ 3y_1 + 5y_2 - 2y_3 - y_4 \leq -7 \\ 3y_1 + 4y_2 - 3y_3 \leq 7 \\ 2y_1 + y_2 - 4y_3 - y_4 \leq 5 \\ 2y_1 + 4y_2 \leq 6 \end{cases}$$

and $y_i \geq 0$ for $i = 1, 2, 3$, and 4 .

Application: A Transportation Problem

A certain company has two factories, F1 and F2, each producing two products, P1 and P2, that are to be shipped to three distribution centers, D1, D2, and D3. The following table illustrates the cost associated with shipping each product from the factory to the distribution center, the minimum number of each product each distribution center needs, and the maximum output of each factory. How much of each product should be shipped from each plant to each distribution center to minimize the total shipping costs?

Solution Let x_1 denote the number of units of P1 shipped from F1 to D1; x_2 the number of units of P2 shipped from F1 to D1; x_3 the number of units of P1 shipped from F1 to D2; x_4 the number of units of P2 shipped from F1 to D2; x_5 the number of units of P1 shipped from F1 to D3; x_6 the number of units of P2 shipped from F1 to D3; x_7 the number of units of P1 shipped from F2 to D1; x_8 the number of units of P2 shipped from F2 to D1; x_9 the number of units of P1 shipped from F2 to D2; x_{10} the number of units of P2 shipped from F2 to D2; x_{11} the number

	F1/P1	F1/P2	F2/P1	F2/P2	Minimum
D1/P1	\$0.75		\$0.80		500
D1/P2		\$0.50		\$0.40	400
D2/P1	\$1.00		\$0.90		300
D2/P2		\$0.75		\$1.20	500
D3/P1	\$0.90		\$0.85		700
D3/P2		\$0.80		\$0.95	300
Maximum output	1000	400	800	900	

of units of P1 shipped from F2 to D3; and x_{12} the number of units of P2 shipped from F2 to D3.

Then, it is necessary to minimize the number

$$Z = .75x_1 + .5x_2 + x_3 + .75x_4 + .9x_5 + .8x_6 + .8x_7 \\ + .4x_8 + .9x_9 + 1.2x_{10} + .85x_{11} + .95x_{12}$$

subject to the constraints $x_1 + x_3 + x_5 \leq 1000$, $x_2 + x_4 + x_6 \leq 400$, $x_7 + x_9 + x_{11} \leq 800$, $x_8 + x_{10} + x_{12} \leq 900$, $x_1 + x_7 \geq 500$, $x_3 + x_9 \geq 500$, $x_5 + x_{11} \geq 700$, $x_2 + x_8 \geq 400$, $x_4 + x_{10} \geq 500$, $x_6 + x_{12} \geq 300$, and x_i nonnegative for $i = 1, 2, \dots, 12$. In order to solve this linear programming problem, the objective function which computes the total cost, the 12 variables, and the set of inequalities must be entered. The coefficients of the objective function are given in the vector **c**. Using the command `Array[x,12]` illustrated in the previous example to define the list of 12 variables $\{x[1], x[2], \dots, x[12]\}$, the objective function is given by the product $z = \mathbf{xvec} \cdot \mathbf{c}$, where **xvec** is the name assigned to the list of variables.

```
Clear[xvec, z, constraints, vars, c]
c = {0.75, 0.5, 1, 0.75, 0.9, 0.8, 0.8, 0.4, 0.9, 1.2,
     0.85, 0.95};
xvec = Array[x, 12]
{x[1], x[2], x[3], x[4], x[5], x[6],
 x[7], x[8], x[9], x[10], x[11], x[12]}

z = xvec.c
0.75x[1] + 0.5x[2] + x[3] + 0.75x[4] +
0.9x[5] + 0.8x[6] + 0.8x[7] + 0.4x[8] +
0.9x[9] + 1.2x[10] + 0.85x[11] + 0.95x[12]
```

The set of constraints are then entered and named **constraints** for easier use. Therefore, the minimum cost and the value of each variable that yields this minimum cost are found with the command

```
Minimize[{z,constraints},xvec]

constraints = {x[1] + x[3] + x[5] ≤ 1000, x[2] + x[4] + x[6] ≤ 400,
  x[7] + x[9] + x[11] ≤ 800, x[8] + x[10] + x[12] ≤ 900,
  x[1] + x[7] ≥ 500, x[3] + x[9] ≥ 300, x[5] + x[11] ≥ 700,
  x[2] + x[8] ≥ 400, x[4] + x[10] > 500, x[6] + x[12] > 300,
  x[1] ≥ 0, x[2] ≥ 0, x[3] ≥ 0, x[4] ≥ 0,
  x[5] ≥ 0, x[6] ≥ 0, x[7] ≥ 0, x[8] ≥ 0,
  x[9] ≥ 0, x[10] ≥ 0, x[11] ≥ 0, x[12] ≥ 0};
values = Minimize[{z, constraints}, xvec]
{2115., {x[1] → 500., x[2] → 0., x[3] → 0., x[4] → 400.,
  x[5] → 200., x[6] → 0., x[7] → 0., x[8] → 400.,
  x[9] → 300., x[10] → 100., x[11] → 500., x[12] → 300.}}
```

Notice that **values** is a list consisting of two elements: the minimum value of the cost function, **2115**, and the list of the variable values $\{x[1] \rightarrow 500, x[2] \rightarrow 0, \dots\}$. Hence, the minimum cost is obtained with the command **values[[1]]** and the list of variable values that yield the minimum cost is extracted with **values[[2]]**.

```
values[[1]]
2115.
values[[2]]
{x[1] → 500., x[2] → 0., x[3] → 0., x[4] → 400.,
  x[5] → 200., x[6] → 0., x[7] → 0., x[8] → 400.,
  x[9] → 300., x[10] → 100., x[11] → 500., x[12] → 300.}
```

Using these extraction techniques, the number of units produced by each factory can be computed. Because x_1 denotes the number of units of P1 shipped from F1 to D1, x_3 the number of units of P1 shipped from F1 to D2, and x_5 the number of units of P1 shipped from F1 to D3, the total number of units of Product 1 produced by Factory 1 is given by the command **x[1]+x[3]+x[5] /. values[[2]]**, which evaluates this sum at the values of **x[1]**, **x[3]**, and **x[5]** given in the list **values[[2]]**.

```
x[1] + x[3] + x[5] /. values[[2]]
700.
```

Also, the number of units of Products 1 and 2 received by each distribution center can be computed. The command **x[3] + x[9] /. values[[2]]** gives the total amount of P1 received at D2 because **x[3]** = amount of P1 received by D2 from F1 and **x[9]** = amount of P1 received by D2 from F2. Notice that this amount is the minimum number of units (300) of P1 requested by D2.

```
x[3] + x[9]/.values[[2]]
300.
```

The number of units of each product that each factory produces can be calculated, and the amount of P1 and P2 received at each distribution center is calculated in a similar manner.

```
{x[1] + x[3] + x[5], x[2] + x[4] + x[6], x[7] + x[9] + x[11],
 x[8] + x[10] + x[12], x[1] + x[7], x[3] + x[9],
 x[5] + x[11], x[2] + x[8],
 x[4] + x[10], x[6] + x[12]}/.values[[2]]//
TableForm
700.
400.
800.
800.
500.
300.
700.
400.
500.
300.
```

From these results, we see that F1 produces 700 units of P1, F1 produces 400 units of P2, F2 produces 800 units of P1, F2 produces 800 units of P2, and each distribution center receives exactly the minimum number of each product it requests.

5.5 SELECTED TOPICS FROM VECTOR CALCULUS

5.5.1 Vector-Valued Functions

Basic operations on two- and three-dimensional vectors are discussed in Section 5.1.4.

We now turn our attention to vector-valued functions. In particular, we consider vector-valued functions of the following forms.

$$\text{Plane curves: } \mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j} \quad (5.3)$$

$$\text{Space curves: } \mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j} + z(t)\mathbf{k} \quad (5.4)$$

$$\text{Parametric surfaces: } \mathbf{r}(s, t) = x(s, t)\mathbf{i} + y(s, t)\mathbf{j} + z(s, t)\mathbf{k} \quad (5.5)$$

$$\text{Vector fields in the plane: } \mathbf{F}(x, y) = P(x, y)\mathbf{i} + Q(x, y)\mathbf{j} \quad (5.6)$$

$$\begin{aligned} \text{Vector fields in space: } \mathbf{F}(x, y, z) = & P(x, y, z)\mathbf{i} \\ & + Q(x, y, z)\mathbf{j} + R(x, y, z)\mathbf{k} \end{aligned} \quad (5.7)$$

For the vector-valued functions (5.3) and (5.4), differentiation and integration are carried out term-by-term, provided that all the terms are differentiable and integrable. Suppose that C is a smooth curve defined by $\mathbf{r}(t)$, $a \leq t \leq b$.

In 2-space, $\mathbf{i} = \langle 1, 0 \rangle$ and $\mathbf{j} = \langle 0, 1 \rangle$. In 3-space, $\mathbf{i} = \langle 1, 0, 0 \rangle$, $\mathbf{j} = \langle 0, 1, 0 \rangle$, and $\mathbf{k} = \langle 0, 0, 1 \rangle$.

1. If $\mathbf{r}'(t) \neq 0$, the **unit tangent vector**, $\mathbf{T}(t)$, is $\mathbf{T}(t) = \frac{\mathbf{r}'(t)}{\|\mathbf{r}'(t)\|}$.
2. If $\mathbf{T}'(t) \neq 0$, the **principal unit normal vector**, $\mathbf{N}(t)$, is $\mathbf{N}(t) = \frac{\mathbf{T}'(t)}{\|\mathbf{T}'(t)\|}$.
3. The **arc length function**, $s(t)$, is $s(t) = \int_a^t \|\mathbf{r}'(u)\| du$. In particular, the length of C on the interval $[a, b]$ is $\int_a^b \|\mathbf{r}'(t)\| dt$.
4. The **curvature**, κ , of C is

$$\kappa = \frac{\|\mathbf{T}'(t)\|}{\|\mathbf{r}'(t)\|} = \frac{\mathbf{a}(t) \cdot \mathbf{N}(t)}{\|\mathbf{v}(t)\|^2} = \frac{\|\mathbf{r}'(t) \times \mathbf{r}''(t)\|}{\|\mathbf{r}'(t)\|^3},$$

where $\mathbf{v}(t) = \mathbf{r}'(t)$ and $\mathbf{a}(t) = \mathbf{r}''(t)$.

It is a good exercise to show that the curvature of a circle of radius r is $1/r$.

Example 5.5.1 (Folium of Descartes). Consider the **folium of Descartes**,

$$\mathbf{r}(t) = \frac{3at}{1+t^3} \mathbf{i} + \frac{3at^2}{1+t^3} \mathbf{j}$$

for $t \neq -1$, if $a = 1$. (a) Find $\mathbf{r}'(t)$, $\mathbf{r}''(t)$, and $\int \mathbf{r}(t) dt$. (b) Find $\mathbf{T}(t)$ and $\mathbf{N}(t)$. (c) Find the curvature, κ . (d) Find the length of the loop of the folium.

Solution (a) After defining $\mathbf{r}(t)$,

$$\mathbf{r}[t] = \{3at/(1+t^3), 3at^2/(1+t^3)\};$$

$$\mathbf{a} = \mathbf{1};$$

we compute $\mathbf{r}'(t)$ and $\int \mathbf{r}(t) dt$ with `'`, `''` and `Integrate`, respectively. We name $\mathbf{r}'(t)$ `dr`, $\mathbf{r}''(t)$ `dr2`, and $\int \mathbf{r}(t) dt `ir`.$

`dr = Simplify[r'[t]]`

`dr2 = Simplify[r''[t]]`

`ir = Integrate[r[t], t]`

$$\left\{ \frac{3-6t^3}{(1+t^3)^2}, -\frac{3t(-2+t^3)}{(1+t^3)^2} \right\}$$

$$\left\{ \frac{18t^2(-2+t^3)}{(1+t^3)^3}, \frac{6(1-7t^3+t^6)}{(1+t^3)^3} \right\}$$

$$\left\{ 3 \left(\frac{\text{ArcTan}\left[\frac{-1+2t}{\sqrt{3}}\right]}{\sqrt{3}} - \frac{1}{3} \text{Log}[1+t] + \frac{1}{6} \text{Log}[1-t+t^2] \right) \right.$$

$$\left. \text{Log}[1+t^3] \right\}$$

(b) Mathematica does not automatically make assumptions regarding the value of t , so it does not algebraically simplify $\|\mathbf{r}'(t)\|$ as we might typically do unless we use `PowerExpand`

`PowerExpand`
`[Sqrt[x^2]]` returns x .

nr = PowerExpand[Sqrt[dr.dr]/Simplify]

$$\frac{3\sqrt{1+4t^2-4t^3-4t^5+4t^6+t^8}}{(1+t^3)^2}$$

The unit tangent vector, $\mathbf{T}(t)$, is formed in `ut`.

ut = dr/nr//Simplify

$$\left\{ \frac{1-2t^3}{\sqrt{1+4t^2-4t^3-4t^5+4t^6+t^8}}, -\frac{t(-2+t^3)}{\sqrt{1+4t^2-4t^3-4t^5+4t^6+t^8}} \right\}$$

We perform the same steps to compute the unit normal vector, $\mathbf{N}(t)$. In particular, note that `dutb = ||T'(t)||`.

dut = D[ut, t]/Simplify

$$\left\{ \frac{2t(-2-3t^3+t^9)}{(1+4t^2-4t^3-4t^5+4t^6+t^8)^{3/2}}, \frac{2-6t^6-4t^9}{(1+4t^2-4t^3-4t^5+4t^6+t^8)^{3/2}} \right\}$$

duta = dut . dut//Simplify

$$\frac{4(1+t^3)^4}{(1+4t^2-4t^3-4t^5+4t^6+t^8)^2}$$

dutb = PowerExpand[Sqrt[duta]]

$$\frac{2(1+t^3)^2}{1+4t^2-4t^3-4t^5+4t^6+t^8}$$

nt = dut/dutb//Simplify

$$\left\{ \frac{t(-2+t^3)}{\sqrt{1+4t^2-4t^3-4t^5+4t^6+t^8}}, \frac{1-2t^3}{\sqrt{1+4t^2-4t^3-4t^5+4t^6+t^8}} \right\}$$

(c) We use the formula $\kappa = \frac{\|\mathbf{T}'(t)\|}{\|\mathbf{r}'(t)\|}$ to determine the curvature in `curvature`.

curvature = Simplify[dutb/nr]

$$\frac{2(1+t^3)^4}{3(1+4t^2-4t^3-4t^5+4t^6+t^8)^{3/2}}$$

We graphically illustrate the unit tangent and normal vectors at $\mathbf{r}(1) = \langle 3/2, 3/2 \rangle$. First, we compute the unit tangent and normal vectors if $t = 1$ using `/.` (`ReplaceAll`).

ut1 = ut/.t -> 1

nt1 = nt/.t -> 1

$$\left\{ -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\}$$

$$\left\{ -\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right\}$$

We then compute the curvature if $t = 1$ in `smallk`. The center of the osculating circle at $\mathbf{r}(1)$ is found in `x0` and `y0`.

The radius of the osculating circle is $1/\kappa$; the position vector of the center is $\mathbf{r} + \frac{1}{\kappa}\mathbf{N}$.

`Graphics[Circle[{x0, y0}, r]]` is a two-dimensional graphics object that represents a circle of radius r centered at the point (x_0, y_0) . Use `Show` to display the graph.

```
smallk = curvature/.t → 1
N[smallk]
x0 = (r[t] + 1 / curvature/.t → 1)[[1]]
y0 = (r[t] + 1 / curvature/.t → 1)[[2]]
 $\frac{8\sqrt{2}}{3}$ 
3.77124
 $\frac{21}{16}$ 
 $\frac{21}{16}$ 
```

We now graph $\mathbf{r}(t)$ with `ParametricPlot`. The unit tangent and normal vectors at $\mathbf{r}(1)$ are graphed with `Arrow` in `a1` and `a2`. The osculating circle at $\mathbf{r}(1)$ is graphed with `Circle` in `c1`. All four graphs are displayed together with `Show` in Figure 5.7.

```
p1 = ParametricPlot[r[t], {t, -100, 100},
PlotRange → {{-2, 3}, {-2, 3}}, AspectRatio → 1];
```

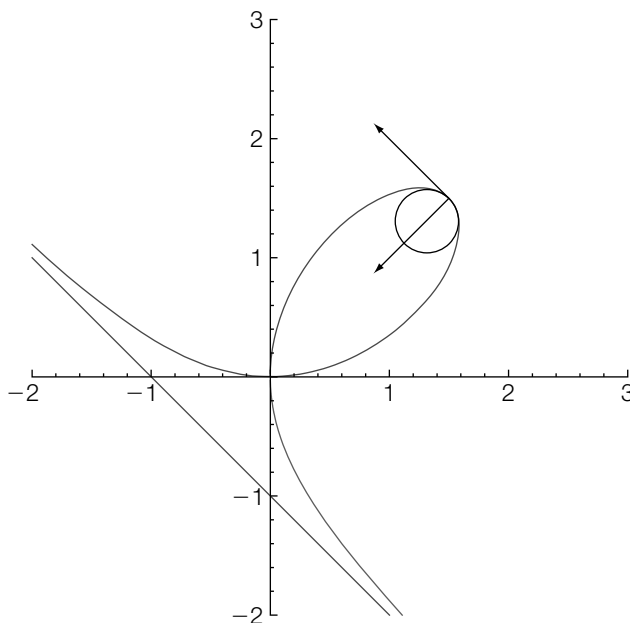


FIGURE 5.7

The folium with an osculating circle

```

p2 = Graphics[{Circle[{x0, y0}, 1/smallk],
  Arrow[{r[1], r[1] + ut1}], Arrow[{r[1], r[1] + nt1}]]];
Show[p1, p2]

```

(d) The loop is formed by graphing $\mathbf{r}(t)$ for $t \geq 0$. Hence, the length of the loop is given by the improper integral $\int_0^\infty \|\mathbf{r}(t)\| dt$, which we compute with `NIntegrate`.

```

NIntegrate[nr, {t, 0, Infinity}]
4.91749

```

In the example, we computed the curvature at $t = 1$. Of course, we could choose other t values. With `Manipulate`,

```

Manipulate[
  r[t_] = {3t/(1 + t^3), 3t^2/(1 + t^3)};
  dr = Simplify[r'[t]];
  dr2 = Simplify[r''[t]];
  ir = Integrate[r[t], t];
  nr = PowerExpand[Sqrt[dr . dr]//Simplify];
  ut = dr/nr//Simplify;
  dut = D[ut, t]//Simplify;
  duta = dut.dut//Simplify;
  dutb = PowerExpand[Sqrt[duta]];
  nt = dut/dutb//Simplify;
  curvature = Simplify[dutb/nr];
  ut1 = ut/.t -> t0;
  nt1 = nt/.t -> t0;
  smallk = curvature/.t -> t0;
  x0 = (r[t] + 1/curvature/.t -> t0)[[1]];
  y0 = (r[t] + 1/curvature/.t -> t0)[[2]];
  p1 = ParametricPlot[r[t], {t, -10, 10},
    PlotRange -> {{-2, 3}, {-2, 3}}, AspectRatio -> 1, PlotPoints -> 200];
  p2 = Graphics[{Circle[{x0, y0}, 1/smallk],
    Arrow[{r[t0], r[t0] + ut1}], Arrow[{r[t0], r[t0] + nt1}]]];
  Show[p1, p2], {{t0, 1}, -5, 10}]

```

we can see the osculating circle at various values of $y_0 \neq -1$. See Figure 5.8(a).

Of course, this particular choice of using the folium to illustrate the procedure could be modified as well. With

```

Manipulate[
  folium[t_] = {3t/(1 + t^3), 3t^2/(1 + t^3)};
  cycloid[t_] = {1/(2Pi)(t - Sin[t]), (1 - Cos[t])/(2Pi)};
  rose[t_] = {3/2Cos[2t]Cos[t], 3/2Cos[2t]Sin[t]};
  squiggle[t_] = {Cos[t] - Sin[2t], Sin[2t] + Cos[5t]};
  cornu[t_] = {2.5FresnelC[t], 2.5FresnelS[t]};
  lissajous[t_] = {2Cos[t], Sin[2t]};

```

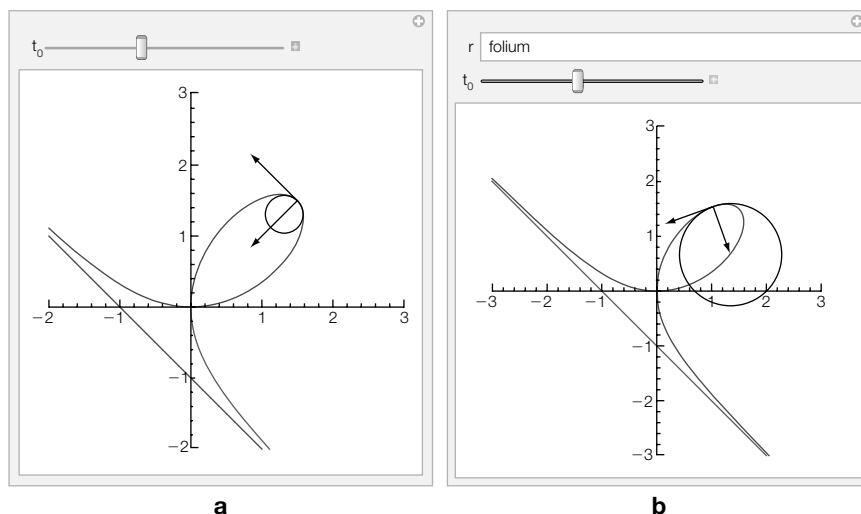


FIGURE 5.8

(a) Using Manipulate to see the osculating circle at various values of t_0 . (b) The osculating circle for various $r(t)$ and t_0

```

evolute[t_] = {Cos[t]^3, 2Sin[t]^3};
dr = Simplify[r'[t]];
dr2 = Simplify[r''[t]];
ir = Integrate[r[t], t];
nr = PowerExpand[Sqrt[dr . dr]]//Simplify;
ut = dr/nr//Simplify;
dut = D[ut, t]//Simplify;
duta = dut . dut//Simplify;
dutb = PowerExpand[Sqrt[duta]];
nt = dut/dutb//Simplify;
curvature = Simplify[dutb/nr];
ut1 = ut/.t -> t0;
nt1 = nt/.t -> t0;
smallk = curvature/.t -> t0;
x0 = (r[t] + 1/curvature/.t -> t0) [[1]];
y0 = (r[t] + 1/curvature/.t -> t0) [[2]];
p1 = ParametricPlot[r[t], {t, -10, 10},
  PlotRange -> {{-3, 3}, {-3, 3}}, AspectRatio -> 1, PlotPoints -> 200];
p2 = Graphics[{Circle[{x0, y0}, 1/smallk],
  Arrow[{r[t0], r[t0] + ut1}], Arrow[{r[t0], r[t0] + nt1}]}];
Show[p1, p2], {{r, folium},
  {folium, cycloid, rose, squiggle, cornu, lissajous, evolute}},
  {{t0, 3/2}, -5, 10}]

```

we allow not only t_0 but also $\mathbf{r}(t)$ to vary. Note that the resulting **Manipulate** object is quite slow on all except the fastest computers. See Figure 5.8(b).

Recall that the **gradient** of $z = f(x, y)$ is the vector-valued function $\nabla f(x, y) = \langle f_x(x, y), f_y(x, y) \rangle$. Similarly, we define the **gradient** of $w = f(x, y, z)$ to be

$$\nabla f(x, y, z) = \langle f_x(x, y, z), f_y(x, y, z), f_z(x, y, z) \rangle = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} + \frac{\partial f}{\partial z} \mathbf{k}. \quad (5.8)$$

A vector field $\mathbf{F}$ is **conservative** if there is a function f , called a **potential function**, satisfying $\nabla f = \mathbf{F}$. In the special case that $\mathbf{F}(x, y) = P(x, y)\mathbf{i} + Q(x, y)\mathbf{j}$, $\mathbf{F}$ is conservative if and only if

$$\frac{\partial P}{\partial y} = \frac{\partial Q}{\partial x}.$$

The **divergence** of the vector field $\mathbf{F}(x, y, z) = P(x, y, z)\mathbf{i} + Q(x, y, z)\mathbf{j} + R(x, y, z)\mathbf{k}$ is the scalar field

$$\operatorname{div} \mathbf{F} = \nabla \cdot \mathbf{F} = \frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} + \frac{\partial R}{\partial z}. \quad (5.9)$$

The **Div** command, which is contained in the **VectorAnalysis** package, can be used to find the divergence of a vector field:

Div[{P(x,y,z),Q(x,y,z),R(x,y,z)},Cartesian[x,y,z]]

computes the divergence of $\mathbf{F}(x, y, z) = P(x, y, z)\mathbf{i} + Q(x, y, z)\mathbf{j} + R(x, y, z)\mathbf{k}$. The **laplacian** of the scalar field $w = f(x, y, z)$ is defined to be

$$\operatorname{div}(\nabla f) = \nabla \cdot (\nabla f) = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = \Delta f. \quad (5.10)$$

In the same way that **Div** computes the divergence of a vector field, **Laplacian**, which is also contained in the **VectorAnalysis** package, computes the laplacian of a scalar field.

The **curl** of the vector field $\mathbf{F}(x, y, z) = P(x, y, z)\mathbf{i} + Q(x, y, z)\mathbf{j} + R(x, y, z)\mathbf{k}$ is

$$\begin{aligned} \operatorname{curl} \mathbf{F}(x, y, z) &= \nabla \times \mathbf{F}(x, y, z) \\ &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ P(x, y, z) & Q(x, y, z) & R(x, y, z) \end{vmatrix} \\ &= \left(\frac{\partial R}{\partial y} - \frac{\partial Q}{\partial z} \right) \mathbf{i} - \left(\frac{\partial R}{\partial x} - \frac{\partial P}{\partial z} \right) \mathbf{j} + \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) \mathbf{k}. \end{aligned} \quad (5.11)$$

If $\mathbf{F}(x, y, z) = P(x, y, z)\mathbf{i} + Q(x, y, z)\mathbf{j} + R(x, y, z)\mathbf{k}$, $\mathbf{F}$ is conservative if and only if $\operatorname{curl} \mathbf{F}(x, y, z) = \mathbf{0}$, in which case $\mathbf{F}$ is said to be **irrotational**.

Example 5.5.2 Determine if

$$\mathbf{F}(x, y) = (1 - 2x^2)ye^{-x^2-y^2}\mathbf{i} + (1 - 2y^2)xe^{-x^2-y^2}\mathbf{j}$$

is conservative. If $\mathbf{F}$ is conservative, find a potential function for $\mathbf{F}$.

Solution We define $P(x, y) = (1 - 2x^2)ye^{-x^2-y^2}$ and $Q(x, y) = (1 - 2y^2)xe^{-x^2-y^2}$. Then we use **D** and **Simplify** to see that $P_y(x, y) = Q_x(x, y)$. Hence, $\mathbf{F}$ is conservative.

```
p[x_, y_] = (1 - 2x^2)y Exp[-x^2 - y^2];
q[x_, y_] = (1 - 2y^2)x Exp[-x^2 - y^2];
Simplify[D[p[x, y], y]]
Simplify[D[q[x, y], x]]
e^{-x^2-y^2} (-1 + 2x^2) (-1 + 2y^2)
e^{-x^2-y^2} (-1 + 2x^2) (-1 + 2y^2)
```

We use **Integrate** to find f satisfying $\nabla f = \mathbf{F}$.

```
i1 = Integrate[p[x, y], x] + g[y]
e^{-x^2-y^2} xy + g[y]

Solve[D[i1, y] == q[x, y], g'[y]]
{{g'[y] -> 0}}
```

Therefore, $g(y) = C$, where C is an arbitrary constant. Letting $C = 0$ gives us the following potential function.

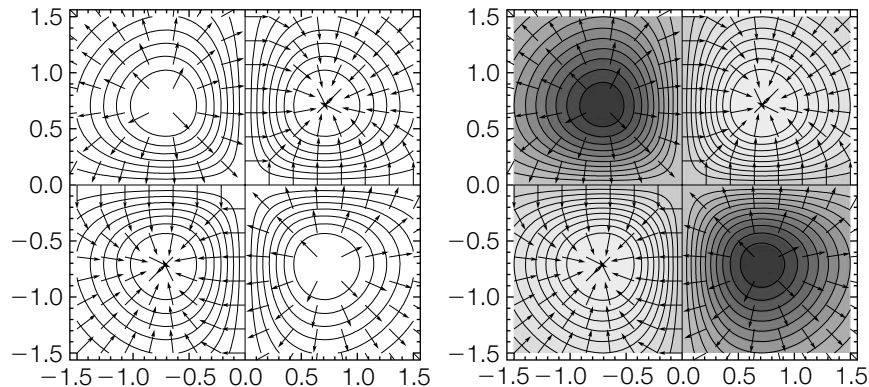
```
f = i1/.g[y] -> 0
e^{-x^2-y^2} xy
```

Remember that the vectors $\mathbf{F}$ are perpendicular to the level curves of f . To see this, we normalize $\mathbf{F}$ in **uv**.

```
uv = {p[x, y], q[x, y]}/
Sqrt[{p[x, y], q[x, y]} . {p[x, y], q[x, y]}}//Simplify
{
  e^{-x^2-y^2} (y - 2x^2 y)
  /
  Sqrt[e^{-2(x^2+y^2)} (y^2 + 4x^4 y^2 + x^2 (1 - 8y^2 + 4y^4))],
  e^{-x^2-y^2} (x - 2xy^2)
  /
  Sqrt[e^{-2(x^2+y^2)} (y^2 + 4x^4 y^2 + x^2 (1 - 8y^2 + 4y^4))]}
}
```

We then graph several level curves of f in **cp1** and **cp2** with **ContourPlot** and several vectors of **uv** with **VectorFieldPlot**, which is contained in the **VectorFieldPlots** package, in **fp**. We show the graphs together with **Show** in Figure 5.9.

```
<<"VectorFieldPlots`"
cp1 = ContourPlot[f, {x, -3/2, 3/2}, {y, -3/2, 3/2}, Contours -> 15,
  ContourShading -> False, PlotPoints -> 60];
```

**FIGURE 5.9**

Two different views illustrating that the vectors $\mathbf{F}$ are perpendicular to the level curves of f

```
cp2 = ContourPlot[f, {x, -3/2, 3/2}, {y, -3/2, 3/2}, Contours -> 20,
  PlotPoints -> 60];
fp = VectorFieldPlot[uv, {x, -3/2, 3/2}, {y, -3/2, 3/2}];
Show[GraphicsRow[{Show[cp1, fp], Show[cp2, fp]}]]
```

Note that we can use `GradientFieldPlot3D`, which is contained in the `VectorFieldPlots` package, to graph several vectors of ∇f . However, the vectors are scaled and it can be difficult to see that the vectors are perpendicular to the level curves of f . The advantage of proceeding this way is that by graphing unit vectors, it is easier to see that the vectors are perpendicular to the level curves of f in the resulting plot.

Example 5.5.3 (a) Show that

$$\mathbf{F}(x, y, z) = -10xy^2\mathbf{i} + (3z^3 - 10x^2y)\mathbf{j} + 9yz^2\mathbf{k}$$

is irrotational. (b) Find f satisfying $\nabla f = \mathbf{F}$. (c) Compute $\operatorname{div} \mathbf{F}$ and $\nabla^2 f$.

Solution (a) After defining $\mathbf{F}(x, y, z)$, we use `Curl`, which is contained in the `VectorAnalysis` package, to see that $\operatorname{curl} \mathbf{F}(x, y, z) = \mathbf{0}$.

```
Needs["VectorAnalysis"]
Clear[f, x, y, z]
f[x_, y_, z_] = {-10xy^2, 3z^3 - 10x^2y, 9yz^2}
{-10xy^2, -10x^2y + 3z^3, 9yz^2}

Curl[f[x, y, z]]
{0, 0, 0}
```

(b) We then use `Integrate` to find $w = f(x, y, z)$ satisfying $\nabla f = \mathbf{F}$.

```
i1 = Integrate[f[x, y, z][[1]], x] + g[y, z]  
-5x2y2 + g[y, z]
```

```
i2 = D[i1, y]  
-10x2y + g(1,0)[y, z]
```

```
Solve[i2==f[x, y, z][[2]], g(1,0)[y, z]]  
{ {g(1,0)[y, z] → 3z3 } }
```

```
i3 = Integrate[3z^3, y] + h[z]  
3yz3 + h[z]
```

```
i4 = i1/.g[y, z] ->i3  
-5x2y2 + 3yz3 + h[z]
```

```
Solve[D[i4, z]==f[x, y, z][[3]]]  
{ {h'[z] → 0} }
```

With $b(z) = C$ and $C = 0$, we have $f(x, y, z) = -5x^2y^2 + 3yz^3$.

```
If = -5x2y2 + 3yz3;
```

∇f is orthogonal to the level surfaces of f . To illustrate this, we use `ContourPlot3D` to graph several level surfaces of $w = f(x, y, z)$ for $-10 \leq x \leq 10$, $-10 \leq y \leq 10$, and $-10 \leq z \leq 10$ in `pf`. We then use `GradientFieldPlot3D`, which is contained in the `VectorFieldPlots` package, to graph several vectors in the gradient field of f over the same domain in `gradf`. The two plots are shown together with `Show` in Figure 5.10. In the plot, notice that the vectors appear to be perpendicular to the surface.

```
pf1=ContourPlot3D[If == -5, {x, -10, 10}, {y, -10, 10},  
  {z, -10, 10}, PlotPoints → 40];  
pf2=ContourPlot3D[If == 10, {x, -10, 10}, {y, -10, 10},  
  {z, -10, 10}, PlotPoints → 40, Mesh → None,  
    ContourStyle → Directive[Red, Opacity[0.8],  
      Specularity[White, 10]]];  
pf3=ContourPlot3D[If == 100, {x, -10, 10}, {y, -10, 10},  
  {z, -10, 10}, Mesh → None,  
    ContourStyle → Directive[Red, Opacity[0.5]],  
      PlotPoints → 40];  
pf4=ContourPlot3D[If, {x, -10, 10}, {y, -10, 10}, {z, -10, 10},  
  PlotPoints → 50, Mesh → None,  
    ContourStyle → Directive[Purple, Opacity[0.3],  
      Specularity[White, 30]]];  
Needs["VectorFieldPlots"]  
gf = GradientFieldPlot3D[If, {x, -10, 10}, {y, -10, 10},
```

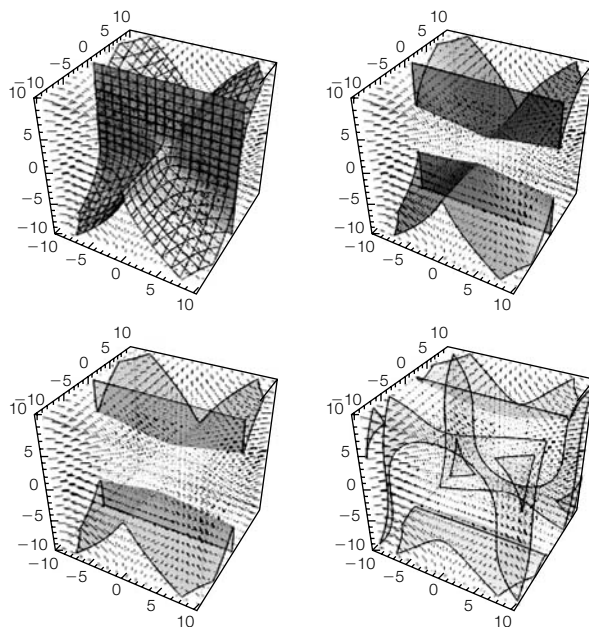


FIGURE 5.10

∇f is orthogonal to the level surfaces of f

```
{z, -10, 10}, PlotPoints -> 15];
Show[GraphicsGrid[{{Show[pf1, gf], Show[pf2, gf]},
{Show[pf3, gf], Show[pf4, gf]}]]]
```

For (c), we take advantage of Div and Laplacian. As expected, the results are the same.

5.5.2 Line Integrals

If $\mathbf{F}$ is continuous on the smooth curve C with parametrization $\mathbf{r}(t)$, $a \leq t \leq b$, the **line integral** of $\mathbf{F}$ on C is

$$\int_C \mathbf{F} \cdot d\mathbf{r} = \int_a^b \mathbf{F} \cdot \mathbf{r}'(t) dt. \quad (5.12)$$

If $\mathbf{F}$ is conservative and C is piecewise smooth, line integrals can be evaluated using the *fundamental theorem of line integrals*.

Theorem 19 (Fundamental Theorem of Line Integrals). *If $\mathbf{F}$ is conservative and the curve C defined by $\mathbf{r}(t)$, $a \leq t \leq b$ is piecewise smooth,*

$$\int_C \mathbf{F} \cdot d\mathbf{r} = f(\mathbf{r}(b)) - f(\mathbf{r}(a)), \quad (5.13)$$

where $\mathbf{F} = \nabla f$.

Example 5.5.4 Find $\int_C \mathbf{F} \cdot d\mathbf{r}$, where $\mathbf{F}(x, y) = (e^{-y} - ye^{-x})\mathbf{i} + (e^{-x} - xe^{-y})\mathbf{j}$ and C is defined by $\mathbf{r}(t) = \cos t\mathbf{i} + \ln(2t/\pi)\mathbf{j}$, $\pi/2 \leq t \leq 4\pi$.

Solution We see that $\mathbf{F}$ is conservative with D and find that $f(x, y) = xe^{-y} + ye^{-x}$ satisfies $\nabla f = \mathbf{F}$ with Integrate.

```
f[x_,y_]={Exp[-y]-y Exp[-x],Exp[-x]-x Exp[-y]};
r[t_]={Cos[t],Log[2t/Pi]};
D[f[x,y][[1]],y]//Simplify
D[f[x,y][[2]],x]//Simplify
-e^-x-e^-y
-e^-x-e^-y

If = Integrate[f[x,y][[1]],x]
e^-yx + e^-xy
```

Hence, using (5.13),

$$\int_C \mathbf{F} \cdot d\mathbf{r} = (xe^{-y} + ye^{-x}) \Big|_{x=0, y=0}^{x=1, y=\ln 8} = \frac{3 \ln 2}{e} + \frac{1}{8} \approx 0.890.$$

```
xr[t_]=Cos[t];
yr[t_]=Log[2t/Pi];
{xr[Pi/2], yr[Pi/2]}
{xr[4Pi], yr[4Pi]}
{0,0}
{1,Log[8]}

Simplify[If/.{x->1, y->Log[8]}]
N[%]
1/8 + Log[8]/e
0.889984
```

If C is a piecewise smooth simple closed curve and $P(x, y)$ and $Q(x, y)$ have continuous partial derivatives, *Green's theorem* relates the line integral $\oint_C (P(x, y) dx + Q(x, y) dy)$ to a double integral.

Theorem 1 (Green's Theorem). *Let C be a piecewise smooth simple closed curve in the plane and R the region bounded by C . If $P(x, y)$ and $Q(x, y)$ have continuous partial derivatives on R ,*

$$\oint_C (P(x, y) dx + Q(x, y) dy) = \iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dA. \quad (5.14)$$

We assume that the symbol $\oint$ means to evaluate the integral in the positive (or counterclockwise) direction.

Example 5.5.5 Evaluate

$$\oint_C (e^{-x} - \sin y) \, dx + (\cos x - e^{-y}) \, dy,$$

where C is the boundary of the region between $y = x^2$ and $x = y^2$.

Solution After defining $P(x, y) = e^{-x} - \sin y$ and $Q(x, y) = \cos x - e^{-y}$, we use `Plot` to determine the region R bounded by C in Figure 5.11.

```
p[x_,y_]=Exp[-x]-Sin[y];
q[x_,y_]=Cos[x]-Exp[-y];
Plot[{x^2,Sqrt[x]},{x,0,1.1},
PlotStyle->{GrayLevel[0],GrayLevel[.3]},
AspectRatio->Automatic]
```

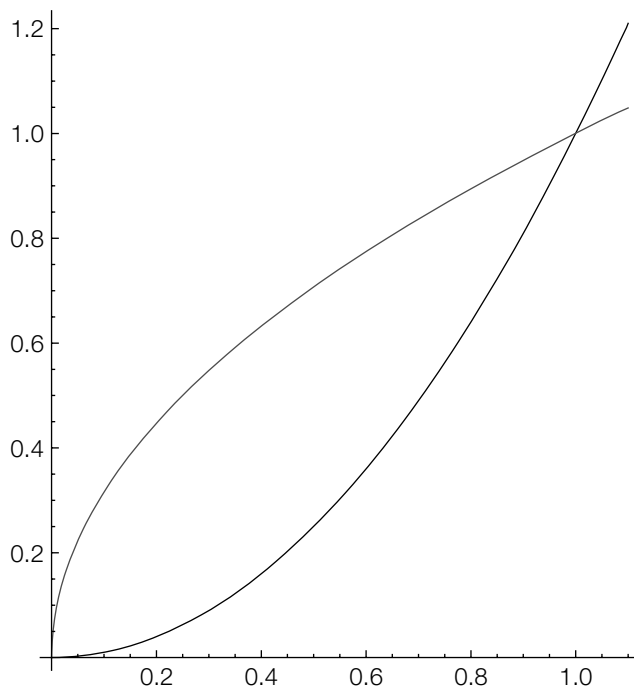


FIGURE 5.11

$y = x^2$ and $y = \sqrt{x}$, $0 \leq x \leq 1$

Using equation (5.14),

$$\begin{aligned}\oint_C (e^{-x} - \sin y) dx + (\cos x - e^{-y}) dy &= \iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dA \\ &= \iint_R (\cos y - \sin x) dA \\ &= \int_0^1 \int_{x^2}^{\sqrt{x}} (\cos y - \sin x) dy dx,\end{aligned}$$

dqdp = Simplify[D[q[x, y], x] - D[p[x, y], y]]

Cos[y] - Sin[x]

which we evaluate with Integrate.

Integrate[dqdp, {x, 0, 1}, {y, x^2, Sqrt[x]}]

N[%]

$$\frac{1}{2} \left(-4 - \sqrt{2\pi} \left(\text{FresnelC} \left[\sqrt{\frac{2}{\pi}} \right] + \text{FresnelS} \left[\sqrt{\frac{2}{\pi}} \right] \right) + 8\text{Sin}[1] \right)$$

0.151091

Notice that the result is given in terms of the **FresnelS** and **FresnelC** functions, which are defined by

$$\text{FresnelS}[x] = \int_0^x \sin \left(\frac{\pi}{2} t^2 \right) dt \quad \text{and} \quad \text{FresnelC}[x] = \int_0^x \cos \left(\frac{\pi}{2} t^2 \right) dt.$$

A more meaningful approximation is obtained with **N**. We conclude that

$$\int_0^1 \int_{x^2}^{\sqrt{x}} (\cos y - \sin x) dy dx \approx 0.151.$$

5.5.3 Surface Integrals

Let S be the graph of $z = f(x, y)$ ($y = b(x, z)$, $x = k(y, z)$) and let R_{xy} , (R_{xz} , R_{yz}) be the projection of S onto the xy (xz , yz) plane. Then,

$$\iint_S g(x, y, z) dS = \iint_{R_{xy}} g(x, y, f(x, y)) \sqrt{[f_x(x, y)]^2 + [f_y(x, y)]^2 + 1} dA \quad (5.15)$$

$$= \iint_{R_{xz}} g(x, b(x, z), z) \sqrt{[b_x(x, z)]^2 + [b_z(x, z)]^2 + 1} dA \quad (5.16)$$

$$= \iint_{R_{yz}} g(k(y, z), y, z) \sqrt{[k_y(y, z)]^2 + [k_z(y, z)]^2 + 1} dA. \quad (5.17)$$

If S is defined parametrically by

$$\mathbf{r}(s, t) = x(s, t)\mathbf{i} + y(s, t)\mathbf{j} + z(s, t)\mathbf{k}, \quad (s, t) \in R,$$

the formula

$$\iint_S g(x, y, z) dS = \iint_R g(\mathbf{r}(s, t)) \|\mathbf{r}_s \times \mathbf{r}_t\| dA, \quad (5.18)$$

where

$$\mathbf{r}_s = \frac{\partial x}{\partial s} \mathbf{i} + \frac{\partial y}{\partial s} \mathbf{j} + \frac{\partial z}{\partial s} \mathbf{k} \quad \text{and} \quad \mathbf{r}_t = \frac{\partial x}{\partial t} \mathbf{i} + \frac{\partial y}{\partial t} \mathbf{j} + \frac{\partial z}{\partial t} \mathbf{k},$$

is also useful.

Theorem 2 (The Divergence Theorem). *Let Q be any domain with the property that each line through any interior point of the domain cuts the boundary in exactly two points, and such that the boundary S is a piecewise smooth closed, oriented surface with unit normal $\mathbf{n}$. If $\mathbf{F}$ is a vector field that has continuous partial derivatives on Q , then*

For our purposes, a surface is **oriented** if it has two distinct sides.

$$\iiint_Q \nabla \cdot \mathbf{F} dV = \iiint_Q \operatorname{div} \mathbf{F} dV = \iint_S \mathbf{F} \cdot \mathbf{n} dS. \quad (5.19)$$

In (5.19), $\iint_S \mathbf{F} \cdot \mathbf{n} dS$ is called the **outward flux** of the vector field $\mathbf{F}$ across the surface S . If S is a portion of the level curve $g(x, y) = C$ for some g , then a unit normal vector $\mathbf{n}$ may be taken to be either

$$\mathbf{n} = \frac{\nabla g}{\|\nabla g\|} \quad \text{or} \quad \mathbf{n} = -\frac{\nabla g}{\|\nabla g\|}.$$

If S is defined parametrically by

$$\mathbf{r}(s, t) = x(s, t)\mathbf{i} + y(s, t)\mathbf{j} + z(s, t)\mathbf{k}, \quad (s, t) \in R,$$

a unit normal vector to the surface is $\mathbf{n} = \frac{\mathbf{r}_s \times \mathbf{r}_t}{\|\mathbf{r}_s \times \mathbf{r}_t\|}$ and (5.19) becomes $\iint_S \mathbf{F} \cdot \mathbf{n} dS = \iint_R \mathbf{F} \cdot (\mathbf{r}_s \times \mathbf{r}_t) dA$.

Example 5.5.6 Find the outward flux of the vector field

$$\mathbf{F}(x, y, z) = (xz + xyz^2) \mathbf{i} + (xy + x^2yz) \mathbf{j} + (yz + xy^2z) \mathbf{k}$$

through the surface of the cube cut from the first octant by the planes $x = 1$, $y = 1$, and $z = 1$.

Solution By the divergence theorem,

$$\iint_{\text{cube surface}} \mathbf{F} \cdot \mathbf{n} dA = \iiint_{\text{cube interior}} \nabla \cdot \mathbf{F} dV.$$

Div is contained in the VectorAnalysis package. You do not need to reload the VectorAnalysis package if you have already loaded it during your *current* Mathematica session.

Hence, without the divergence theorem, calculating the outward flux would require six separate integrals, corresponding to the six faces of the cube. After defining $\mathbf{F}$, we compute $\nabla \cdot \mathbf{F}$ with Div.

Needs["VectorAnalysis"]

f[x_, y_, z_] = {xz + xyz^2, xy + x^2yz, yz + xy^2z};

divf = Div[f[x, y, z], Cartesian[x, y, z]]

x + y + xy^2 + z + x^2z + yz^2

The outward flux is then given by

$$\iiint_{\text{cube interior}} \nabla \cdot \mathbf{F} dV = \int_0^1 \int_0^1 \int_0^1 \nabla \cdot \mathbf{F} dz dy dx = 2,$$

which we compute with Integrate.

Integrate[divf, {z, 0, 1}, {y, 0, 1}, {x, 0, 1}]

2

Theorem 3 (Stokes' Theorem). *Let S be an oriented surface with finite surface area, unit normal $\mathbf{n}$, and boundary C . Let $\mathbf{F}$ be a continuous vector field defined on S such that the components of $\mathbf{F}$ have continuous partial derivatives at each nonboundary point of S . Then,*

$$\oint_C \mathbf{F} \cdot d\mathbf{r} = \iint_S \text{curl } \mathbf{F} \cdot \mathbf{n} dS. \quad (5.20)$$

In other words, the surface integral of the normal component of the curl of $\mathbf{F}$ taken over S equals the line integral of the tangential component of the field taken over C . In particular, if $\mathbf{F} = P(x, y, z)\mathbf{i} + Q(x, y, z)\mathbf{j} + R(x, y, z)\mathbf{k}$, then

$$\int_C (P(x, y, z)dx + Q(x, y, z)dy + R(x, y, z)dz) = \iint_S \text{curl } \mathbf{F} \cdot \mathbf{n} dS.$$

Example 5.5.7 Verify Stokes' theorem for the vector field

$$\mathbf{F}(x, y, z) = (x^2 - y)\mathbf{i} + (y^2 - z)\mathbf{j} + (x + z^2)\mathbf{k}$$

and S the portion of the paraboloid $z = f(x, y) = 9 - (x^2 + y^2)$, $z \geq 0$.

Solution After loading the VectorAnalysis package, we define $\mathbf{F}$ and f . The curl of $\mathbf{F}$ is computed with Curl in curlF.

Needs["VectorAnalysis"]

capf[x_, y_, z_] = {x^2 - y, y^2 - z, x + z^2};

f[x_, y_] = 9 - (x^2 + y^2);

curlcapf = Curl[capf[x, y, z], Cartesian[x, y, z]]

{1, -1, 1}

Next, we define the function $b(x, y, z) = z - f(x, y)$. A normal vector to the surface is given by ∇b . A unit normal vector, $\mathbf{n}$, is then given by $\mathbf{n} = \frac{\nabla b}{\|\nabla b\|}$, which is computed in un.

```
h[x_, y_, z_] = z - f[x, y]
normtosurf = Grad[h[x, y, z], Cartesian[x, y, z]]
-9 + x^2 + y^2 + z
{2x, 2y, 1}

un = Simplify[normtosurf/Sqrt[normtosurf . normtosurf]]
{
   $\frac{2x}{\sqrt{1+4x^2+4y^2}}$ ,  $\frac{2y}{\sqrt{1+4x^2+4y^2}}$ ,  $\frac{1}{\sqrt{1+4x^2+4y^2}}$ 
}
```

The dot product $\text{curl } \mathbf{F} \cdot \mathbf{n}$ is computed in g.

```
g = Simplify[curlcapf . un]
 $\frac{1+2x-2y}{\sqrt{1+4x^2+4y^2}}$ 
```

Using the surface integral evaluation formula (5.15),

In this example, R , the projection of $f(x, y)$ onto the xy -plane, is the region bounded by the graph of the circle $x^2 + y^2 = 9$.

$$\begin{aligned} \iint_S \text{curl } \mathbf{F} \cdot \mathbf{n} \, dS &= \iint_R g(x, y, f(x, y)) \sqrt{[f_x(x, y)]^2 + [f_y(x, y)]^2 + 1} \, dA \\ &= \int_{-3}^3 \int_{-\sqrt{9-x^2}}^{\sqrt{9-x^2}} g(x, y, f(x, y)) \sqrt{[f_x(x, y)]^2 + [f_y(x, y)]^2 + 1} \, dy \, dx \\ &= 9\pi, \end{aligned}$$

which we compute with Integrate.

```
tointegrate =
Simplify[(g/.z->f[x, y]) *
Sqrt[D[f[x, y], x]^2 + D[f[x, y], y]^2 + 1]]
1 + 2x - 2y

i1 = Integrate[tointegrate, {x, -3, 3},
{y, -Sqrt[9 - x^2], Sqrt[9 - x^2]]]
9π
```

To verify Stokes' theorem, we must compute the associated line integral. Notice that the boundary of $z = f(x, y) = 9 - (x^2 + y^2)$, $z = 0$, is the circle $x^2 + y^2 = 9$ with parametrization $x = 3 \cos t$, $y = 3 \sin t$, $z = 0$, $0 \leq t \leq 2\pi$. This parametrization is substituted into $\mathbf{F}(x, y, z)$ and named pvf.

```
pvf = capf[3Cos[t], 3Sin[t], 0]
{9Cos[t]^2 - 3Sin[t], 9Sin[t]^2, 3Cos[t]}
```

To evaluate the line integral along the circle, we next define the parametrization of the circle and calculate $d\mathbf{r}$. The dot product of $\mathbf{pvf}$ and $d\mathbf{r}$ represents the integrand of the line integral.

```
r[t_]= {3Cos[t], 3Sin[t], 0};
dr = r'[t]
{ -3Sin[t], 3Cos[t], 0}
tointegrate = pvf . dr;
```

As before with x and y , we instruct Mathematica to assume that t is real, compute the dot product of $\mathbf{pvf}$ and $d\mathbf{r}$, and evaluate the line integral with `Integrate`.

```
Integrate[tointegrate, {t, 0, 2Pi}]
9π
```

As expected, the result is 9π .

5.5.4 A Note on Nonorientability

See “When is a surface *not* orientable?” by Braselton, Abell, and Braselton [5] for a detailed discussion regarding the examples in this section.

Suppose that S is the surface determined by

$$\mathbf{r}(s, t) = x(s, t)\mathbf{i} + y(s, t)\mathbf{j} + z(s, t)\mathbf{k}, \quad (s, t) \in R$$

and let

$$\mathbf{n} = \frac{\mathbf{r}_s \times \mathbf{r}_t}{\|\mathbf{r}_s \times \mathbf{r}_t\|} \quad \text{or} \quad \mathbf{n} = -\frac{\mathbf{r}_s \times \mathbf{r}_t}{\|\mathbf{r}_s \times \mathbf{r}_t\|}, \quad (5.21)$$

where

$$\mathbf{r}_s = \frac{\partial x}{\partial s}\mathbf{i} + \frac{\partial y}{\partial s}\mathbf{j} + \frac{\partial z}{\partial s}\mathbf{k} \quad \text{and} \quad \mathbf{r}_t = \frac{\partial x}{\partial t}\mathbf{i} + \frac{\partial y}{\partial t}\mathbf{j} + \frac{\partial z}{\partial t}\mathbf{k},$$

if $\|\mathbf{r}_s \times \mathbf{r}_t\| \neq 0$. If $\mathbf{n}$ is defined, $\mathbf{n}$ is orthogonal (or perpendicular) to S . We state three familiar definitions of *orientable*.

- S is **orientable** if S has a unit normal vector field, $\mathbf{n}$, that varies continuously between any two points (x_0, y_0, z_0) and (x_1, y_1, z_1) on S . (See [7].)
- S is **orientable** if S has a continuous unit normal vector field, $\mathbf{n}$. (See [7] and [19].)
- S is **orientable** if a unit vector $\mathbf{n}$ can be defined at every nonboundary point of S in such a way that the normal vectors vary continuously over the surface S . (See [14].)

A path is **order preserving** if our chosen orientation is preserved as we move along the path.

Thus, a surface such as a torus is orientable.

Example 5.5.8 (The Torus). Using the standard parametrization of the torus, we use ParametricPlot3D to plot the torus if $c = 3$ and $a = 1$ in Figure 5.12.

Also see
Example 2.3.18.

```
Clear[r, c, a, x, y, z, r]
c = 3;
a = 1;
x[s_, t_] = (c + aCos[s])Cos[t];
y[s_, t_] = (c + aCos[s])Sin[t];
z[s_, t_] = aSin[s];
r[s_, t_] = {x[s, t], y[s, t], z[s, t]};
threeDpt = ParametricPlot3D[r[s, t], {s, -Pi, Pi},
  {t, -Pi, Pi}, PlotPoints -> {30, 30}, AspectRatio -> 1,
  PlotRange -> {{-4, 4}, {-4, 4}, {-1, 1}},
  BoxRatios -> {4, 4, 1}, AxesLabel -> {"x", "y", "z"},
  ColorFunction -> "FruitPunchColors", PlotStyle -> Opacity[.3]]
```

To plot a normal vector field on the torus, we compute $\frac{\partial}{\partial s} \mathbf{r}(s, t)$,

```
rs = D[r[s, t], s]
{-Cos[t]Sin[s], -Sin[s]Sin[t], Cos[s]}
```

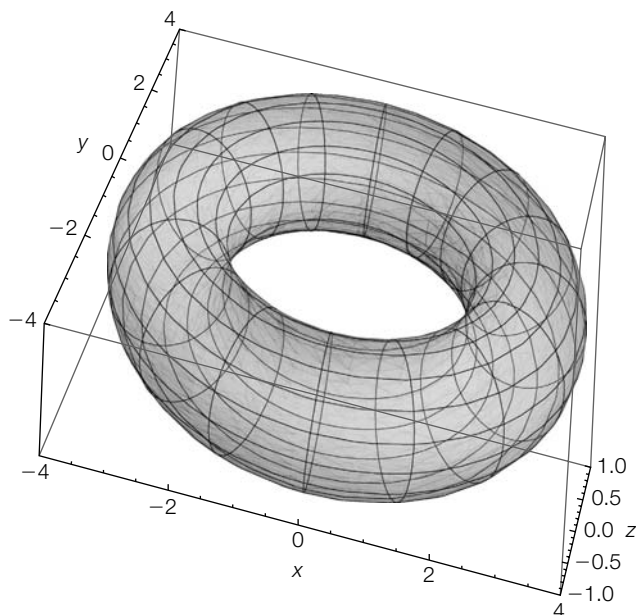


FIGURE 5.12

A torus

and $\frac{\partial}{\partial t} \mathbf{r}(s, t)$.

$$\mathbf{rt} = \mathbf{D}[\mathbf{r}[s, t], t] \\ \{-(3 + \cos[s])\sin[t], (3 + \cos[s])\cos[t], 0\}$$

The cross product $\frac{\partial}{\partial s} \mathbf{r}(s, t) \times \frac{\partial}{\partial t} \mathbf{r}(s, t)$ is formed in `rscrossrt`.

$$\mathbf{rscrossrt} = \mathbf{Cross}[\mathbf{rs}, \mathbf{rt}] // \mathbf{Simplify} \\ \{-\cos[s](3 + \cos[s])\cos[t], \\ -\cos[s](3 + \cos[s])\sin[t], -(3 + \cos[s])\sin[s]\}$$

$$\mathbf{Sqrt}[\mathbf{rscrossrt} . \mathbf{rscrossrt}] // \mathbf{FullSimplify} \\ \sqrt{(3 + \cos[s])^2}$$

Using equation (5.24), we define `un`: Given s and t , `un[s,t]` returns a unit normal to the torus.

$$\mathbf{Clear}[\mathbf{un}] \\ \mathbf{un}[s_ , t_] = \\ -\mathbf{rscrossrt} / \mathbf{Sqrt}[\mathbf{rscrossrt} . \\ \mathbf{rscrossrt}] // \mathbf{PowerExpand} // \mathbf{FullSimplify}$$

$$\left\{ \frac{\cos[s](3 + \cos[s])\cos[t]}{\sqrt{(3 + \cos[s])^2}}, \right. \\ \left. \frac{\cos[s](3 + \cos[s])\sin[t]}{\sqrt{(3 + \cos[s])^2}}, \frac{(3 + \cos[s])\sin[s]}{\sqrt{(3 + \cos[s])^2}} \right\}$$

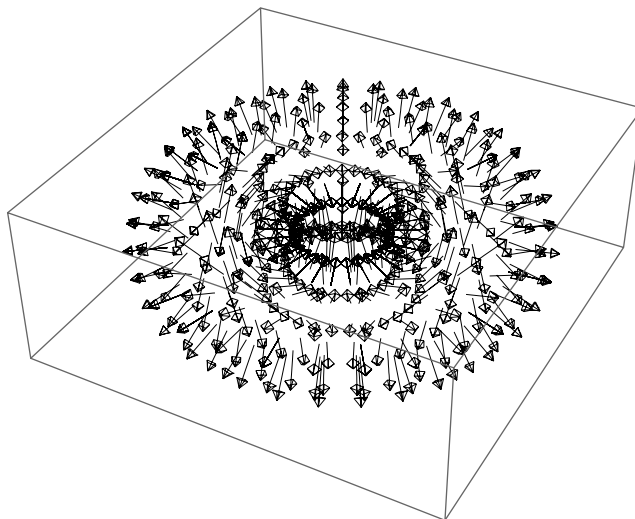
$$\mathbf{Map}[\mathbf{PowerExpand}, \mathbf{un}[s, t]] \\ \{\cos[s]\cos[t], \cos[s]\sin[t], \sin[s]\}$$

$$\mathbf{r}[s, t] \\ \{(3 + \cos[s])\cos[t], \\ (3 + \cos[s])\sin[t], \sin[s]\}$$

$$\mathbf{un}[s, t] \\ \left\{ \frac{\cos[s](3 + \cos[s])\cos[t]}{\sqrt{(3 + \cos[s])^2}}, \right. \\ \left. \frac{\cos[s](3 + \cos[s])\sin[t]}{\sqrt{(3 + \cos[s])^2}}, \frac{(3 + \cos[s])\sin[s]}{\sqrt{(3 + \cos[s])^2}} \right\}$$

To plot the normal vector field on the torus, we take advantage of the command `ListVectorFieldPlot3D`, which is contained in the **VectorFieldPlots** package. See Figure 5.13.

```
<<"VectorFieldPlots"
Clear[vecs]
vecs = Flatten[Table[{r[s, t], un[s, t]},
  {s, -Pi, Pi, 2Pi/14}, {t, -Pi, Pi, 2Pi/29}], 1];
pp2 = ListVectorFieldPlot3D[vecs, VectorHeads -> True]
Show[threedp1t, pp2, AspectRatio -> 1,
```

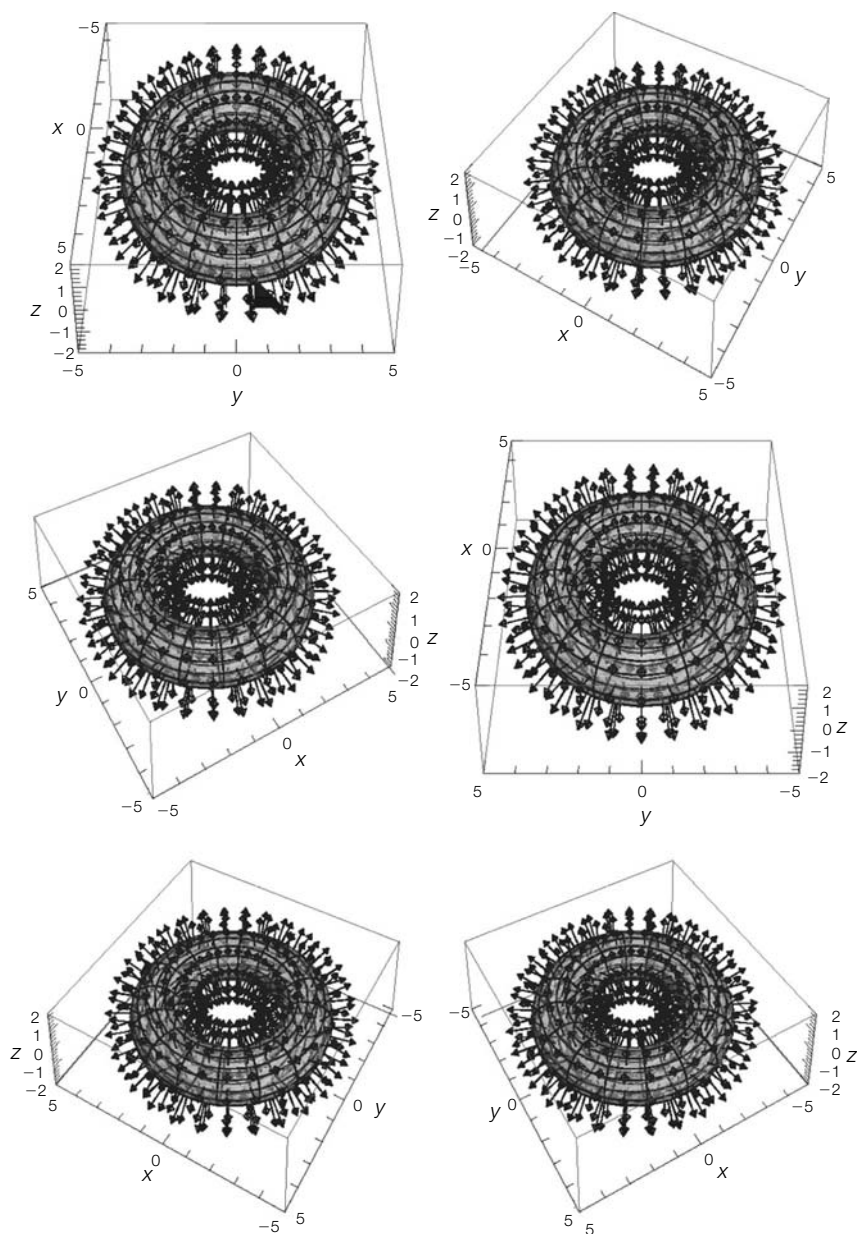
**FIGURE 5.13**

Unit normal vector field on a torus

```
PlotRange -> {{-5, 5}, {-5, 5}, {-2, 2}},
BoxRatios -> {4, 4, 1}, AxesLabel -> {"x", "y", "z"}}
```

We use `Show` (illustrating the use of the `ViewPoint` option) together with `GraphicsArray` to see the vector field on the torus together from various angles in Figure 5.14. Regardless of the viewing angle, the figure looks the same; the torus is orientable.

```
g1 = Show[threedp1t, pp2, AspectRatio -> 1,
PlotRange -> {{-5, 5}, {-5, 5}, {-2, 2}},
BoxRatios -> {4, 4, 1}, AxesLabel -> {"x", "y", "z"},
ViewPoint -> {2.729, -0.000, 2.000}];
g2 = Show[threedp1t, pp2, AspectRatio -> 1,
PlotRange -> {{-5, 5}, {-5, 5}, {-2, 2}},
BoxRatios -> {4, 4, 1}, AxesLabel -> {"x", "y", "z"},
ViewPoint -> {1.365, -2.364, 2.000}];
g3 = Show[threedp1t, pp2, AspectRatio -> 1,
PlotRange -> {{-5, 5}, {-5, 5}, {-2, 2}},
BoxRatios -> {4, 4, 1}, AxesLabel -> {"x", "y", "z"},
ViewPoint -> {-1.365, -2.364, 2.000}];
g4 = Show[threedp1t, pp2, AspectRatio -> 1,
PlotRange -> {{-5, 5}, {-5, 5}, {-2, 2}},
BoxRatios -> {4, 4, 1}, AxesLabel -> {"x", "y", "z"},
ViewPoint -> {-2.729, 0.000, 2.000}];
g5 = Show[threedp1t, pp2, AspectRatio -> 1,
PlotRange -> {{-5, 5}, {-5, 5}, {-2, 2}},
```

**FIGURE 5.14**

The torus is orientable

```

BoxRatios->{4, 4, 1}, AxesLabel->{"x", "y", "z"},
ViewPoint->{-1.365, 2.364, 2.000}];
g6 = Show[threedp1, pp2, AspectRatio->1,
PlotRange->{{-5, 5}, {-5, 5}, {-2, 2}},
BoxRatios->{4, 4, 1}, AxesLabel->{"x", "y", "z"},
ViewPoint->{1.365, 2.364, 2.000}];
Show[GraphicsGrid[{{g1, g2}, {g3, g4}, {g5, g6}}]]

```

If a 2-manifold, S , has an **order reversing path** (or **not order preserving path**), S is **nonorientable** (or **not orientable**).

Determining whether a given surface S is orientable or not may be a difficult problem.

Example 5.5.9 (The Möbius Strip). The *Möbius strip* is frequently cited as an example of a nonorientable surface with boundary: It has one side and is physically easy to construct by hand by half twisting and taping (or pasting) together the ends of a piece of paper (for example, see [5], [7], [14], and [19]). A parametrization of the Möbius strip is $\mathbf{r}(s, t) = x(s, t)\mathbf{i} + y(s, t)\mathbf{j} + z(s, t)\mathbf{k}$, $-1 \leq s \leq 1$, $-\pi \leq t \leq \pi$, where

$$x = \left[c + s \cos \left(\frac{1}{2}t \right) \right] \cos t, \quad y = \left[c + s \cos \left(\frac{1}{2}t \right) \right] \sin t, \quad \text{and} \\ z = s \sin \left(\frac{1}{2}t \right), \quad (5.22)$$

and we assume that $c > 1$. In Figure 5.15, we graph the Möbius strip using $c = 3$.

```

c = 3;
x[s_, t_] = (c + sCos[t/2])Cos[t];
y[s_, t_] = (c + sCos[t/2])Sin[t];
z[s_, t_] = sSin[t/2];
r[s_, t_] = {x[s, t], y[s, t], z[s, t]};
threedp1 = ParametricPlot3D[r[s, t], {s, -1, 1},
{t, -Pi, Pi}, PlotPoints->{30, 30},
AspectRatio->1, PlotRange->
{{-4, 4}, {-4, 4}, {-1, 1}}, BoxRatios->{4, 4, 1},
AxesLabel->{"x", "y", "z"}, ColorFunction->"NeonColors",
Mesh->False, PlotStyle->Opacity[.8]]

```

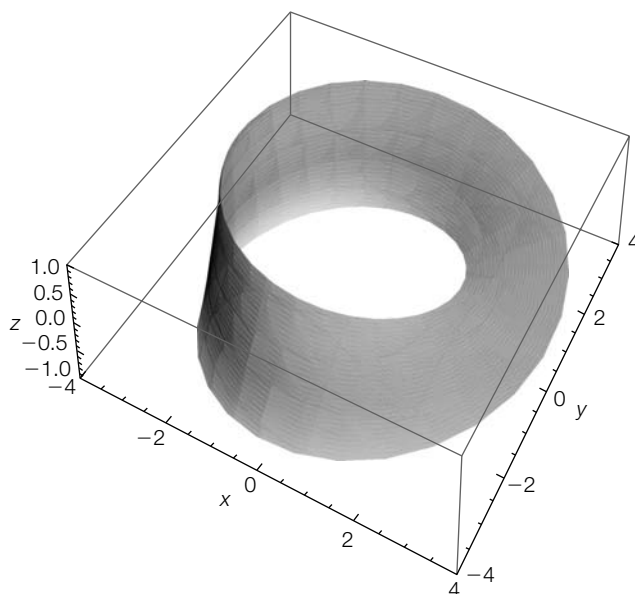
Although it is relatively easy to see in the plot that the Möbius strip has only one side, the fact that a unit vector, $\mathbf{n}$, normal to the Möbius strip at a point P reverses its direction as $\mathbf{n}$ moves around the strip to P is not obvious to the novice.

With Mathematica, we compute $\|\mathbf{r}_s \times \mathbf{r}_t\|$ and $\mathbf{n} = \frac{\mathbf{r}_s \times \mathbf{r}_t}{\|\mathbf{r}_s \times \mathbf{r}_t\|}$.

```

rs = D[r[s, t], s]
{Cos [1/2] Cos[t], Cos [1/2] Sin[t], Sin [1/2]}

```

**FIGURE 5.15**

Parametric plot of equations (5.22) if $c = 3$

rt = D[r[s, t], t]

$$\left\{ -\frac{1}{2}s\cos[t]\sin\left[\frac{t}{2}\right] - \left(3 + s\cos\left[\frac{t}{2}\right]\right)\sin[t], \right. \\ \left. \left(3 + s\cos\left[\frac{t}{2}\right]\right)\cos[t] - \frac{1}{2}s\sin\left[\frac{t}{2}\right]\sin[t], \right. \\ \left. \frac{1}{2}s\cos\left[\frac{t}{2}\right] \right\}$$

rscrossrt = Cross[rs, rt]//Simplify

$$\left\{ -\frac{1}{2}\left(-s\cos\left[\frac{t}{2}\right] + 6\cos[t] + \right. \right. \\ \left. \left. s\cos\left[\frac{3t}{2}\right]\right)\sin\left[\frac{t}{2}\right], \right. \\ \left. \frac{1}{4}\left(-s - 6\cos\left[\frac{t}{2}\right] - 2s\cos[t] + \right. \right. \\ \left. \left. 6\cos\left[\frac{3t}{2}\right] + s\cos[2t]\right), \right. \\ \left. \cos\left[\frac{t}{2}\right]\left(3 + s\cos\left[\frac{t}{2}\right]\right) \right\}$$

Sqrt[rscrossrt . rscrossrt]//FullSimplify

$$\sqrt{9 + \frac{3s^2}{4} + 6s\cos\left[\frac{t}{2}\right] + \frac{1}{2}s^2\cos[t]}$$

Clear[un]

un[s_, t_] =

rscrossrt/Sqrt[rscrossrt . rscrossrt]//FullSimplify

$$\left\{ \frac{s\sin[t] - \cos[t](6\sin\left[\frac{t}{2}\right] + s\sin[t])}{\sqrt{36 + 3s^2 + 24s\cos\left[\frac{t}{2}\right] + 2s^2\cos[t]}}, \right.$$

$$\left. \begin{aligned} & -\frac{3\cos\left[\frac{1}{2}\right] - 3\cos\left[\frac{3t}{2}\right] + s(\cos[t] + \sin[t]^2)}{\sqrt{36 + 3s^2 + 24s\cos\left[\frac{1}{2}\right] + 2s^2\cos[t]}}, \\ & \frac{s + 6\cos\left[\frac{1}{2}\right] + s\cos[t]}{\sqrt{36 + 3s^2 + 24s\cos\left[\frac{1}{2}\right] + 2s^2\cos[t]}} \end{aligned} \right\}$$

Consider the path C given by $\mathbf{r}(0, t)$, $-\pi \leq t \leq \pi$ that begins and ends at $\langle -3, 0, 0 \rangle$. On C , $\mathbf{n}(0, t)$ is given by

un[0, t]

$$\left\{ -\cos[t]\sin\left[\frac{1}{2}\right], \frac{1}{6} \left(-3\cos\left[\frac{1}{2}\right] + 3\cos\left[\frac{3t}{2}\right] \right), \cos\left[\frac{1}{2}\right] \right\}$$

At $t = -\pi$, $\mathbf{n}(0, -\pi) = \langle 1, 0, 0 \rangle$, whereas at $t = \pi$, $\mathbf{n}(0, \pi) = \langle -1, 0, 0 \rangle$.

r[0, -Pi]

r[0, Pi]

$$\{-3, 0, 0\}$$

$$\{\pm 3, 0, 0\}$$

As $\mathbf{n}$ moves along C from $\mathbf{r}(0, -\pi)$ to $\mathbf{r}(0, \pi)$, the orientation of $\mathbf{n}$ reverses, as shown in Figure 5.16.

```
l1 = Table[r[0, t], {t, -Pi, Pi, 2Pi/179}];
threedp2 = Show[Graphics3D[{Thickness[.02],
  GrayLevel[.6], Line[l1]}], Axes -> Automatic,
  PlotRange -> {{-4, 4}, {-4, 4}, {-1, 1}},
  BoxRatios -> {4, 4, 1}, AspectRatio -> 1];
<< "VectorFieldPlots";
```

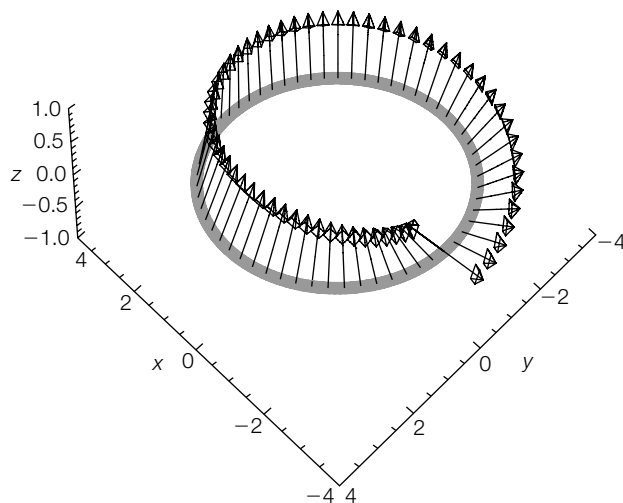


FIGURE 5.16

Parametric plot of equations (5.22) if $c = 3$

```

vecs = Table[{r[0, t], un[0, t]}, {t, - $\pi$ ,  $\pi$ ,  $\frac{2\pi}{59}$ ]];
pp2 = ListVectorFieldPlot3D[vecs, VectorHeads  $\rightarrow$  True];
Show[threedp2, pp2, ViewPoint  $\rightarrow$ 
      {-2.093, 2.124, 1.600}, AxesLabel  $\rightarrow$  {"x", "y", "z"},
      Boxed  $\rightarrow$  False, DisplayFunction  $\rightarrow$  $DisplayFunction]

```

Several different views of Figure 5.16 on the Möbius strip shown in Figure 5.15 are shown in Figure 5.17. C is an orientation reversing path and we can conclude that the Möbius strip is not orientable.

An animation is particularly striking.

```

g1 = Show[threedp1, threedp2, pp2,
      ViewPoint  $\rightarrow$  {2.729, -0.000, 2.000},
      AxesLabel  $\rightarrow$  {"x", "y", "z"}, Boxed  $\rightarrow$  False];
g2 = Show[threedp1, threedp2, pp2,
      ViewPoint  $\rightarrow$  {1.365, -2.364, 2.000},
      AxesLabel  $\rightarrow$  {"x", "y", "z"}, Boxed  $\rightarrow$  False];
g3 = Show[threedp1, threedp2, pp2,
      ViewPoint  $\rightarrow$  {-1.365, -2.364, 2.000},
      AxesLabel  $\rightarrow$  {"x", "y", "z"}, Boxed  $\rightarrow$  False];
g4 = Show[threedp1, threedp2, pp2,
      ViewPoint  $\rightarrow$  {-2.729, 0.000, 2.000},
      AxesLabel  $\rightarrow$  {"x", "y", "z"}, Boxed  $\rightarrow$  False];
g5 = Show[threedp1, threedp2, pp2,
      ViewPoint  $\rightarrow$  {-1.365, 2.364, 2.000},
      AxesLabel  $\rightarrow$  {"x", "y", "z"}, Boxed  $\rightarrow$  False];
g6 = Show[threedp1, threedp2, pp2,
      ViewPoint  $\rightarrow$  {1.365, 2.364, 2.000},
      AxesLabel  $\rightarrow$  {"x", "y", "z"}, Boxed  $\rightarrow$  False];
Show[GraphicsGrid[{{g1, g2}, {g3, g4}, {g5, g6}}]]

```

Example 5.5.10 The *Klein bottle* is an interesting surface with neither an inside nor an outside, which indicates to us that it is not orientable. In Figure 5.18(a) we show the “usual” *immersion* of the Klein bottle. Although the Klein bottle does not intersect itself, it is not possible to visualize it in Euclidean 3-space without it doing so. Visualizations of 2-manifolds such as the Klein bottle’s “usual” rendering in Euclidean 3-space are called *immersions*. (See [11] for a nontechnical discussion of immersions.)

```

r = 4(1 - 1/2Cos[u]);
x1[u_, v_] = 6(1 + Sin[u])Cos[u] + rCos[u]Cos[v];
x2[u_, v_] = 6(1 + Sin[u])Cos[u] + rCos[v + Pi];
y1[u_, v_] = 16Sin[u] + rSin[u]Cos[v];
y2[u_, v_] = 16Sin[u];
z[u_, v_] = rSin[v];
kb1a = ParametricPlot3D[{x1[s, t], y1[s, t], z[s, t]},
      {s, 0, Pi}, {t, 0, 2Pi}, PlotPoints  $\rightarrow$  {30, 30},

```

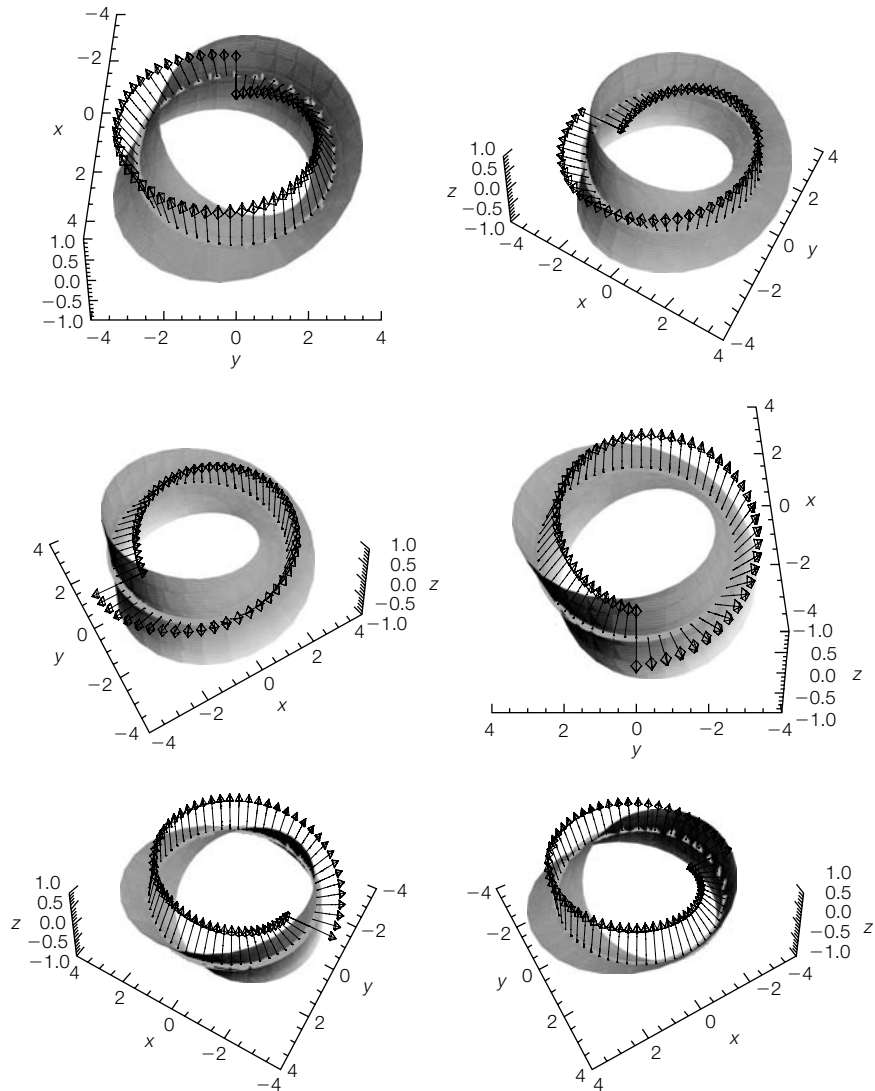


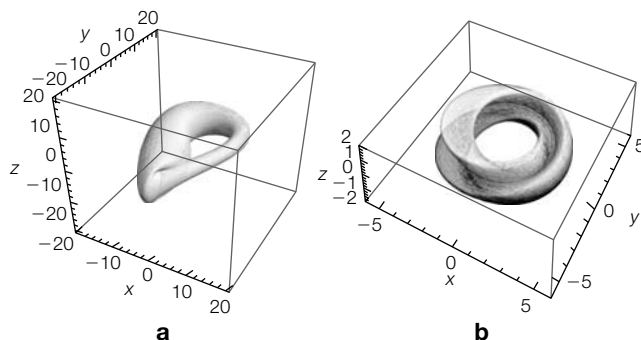
FIGURE 5.17

Different views of a Möbius strip with an orientation reversing path

```

AspectRatio -> 1, AxesLabel -> {"x", "y", "z"},
Mesh -> False, PlotStyle -> Opacity[.8];
kb1b = ParametricPlot3D[{x1[s, t], y1[s, t], z[s, t]},
{s, Pi, 2Pi}, {t, 0, 2Pi}, PlotPoints -> {30, 30},
AspectRatio -> 1, AxesLabel -> {"x", "y", "z"},
Mesh -> False, PlotStyle -> Opacity[.8]
kb1 = Show[kb1a, kb1b, PlotRange -> {{-20, 20}, {-20, 20}, {-20, 20}}]

```

**FIGURE 5.18**

Two different immersions of the Klein bottle: (a) the “usual” immersion; (b) the figure-8 immersion

Figure 5.18(b) shows the *figure-8* immersion of the Klein bottle. Notice that it is not easy to see that the Klein bottle has neither an inside nor an outside in Figure 5.14.

a = 3;

x[u_, v_] = (a + Cos[u/2]Sin[v] - Sin[u/2]Sin[2v])Cos[u];

y[u_, v_] = (a + Cos[u/2]Sin[v] - Sin[u/2]Sin[2v])Sin[u];

z[u_, v_] = Sin[u/2]Sin[v] + Cos[u/2]Sin[2v];

r[u_, v_] = {x[u, v], y[u, v], z[u, v]};

ParametricPlot3D[r[t, t], {t, 0, 2Pi}]

kb2 = ParametricPlot3D[r[s, t], {s, -Pi, Pi}, {t, -Pi, Pi},

PlotPoints -> {30, 30}, AspectRatio -> 1,

AxesLabel -> {"x", "y", "z"},

PlotRange -> {{-6, 6}, {-6, 6}, {-2, 2}}, BoxRatios -> {4, 4, 1},

ColorFunction -> "SunsetColors", Mesh -> False,

PlotStyle -> Opacity[.4]]

Show[GraphicsRow[{kb1, kb2}]]

In fact, to many readers it may not be clear whether the Klein bottle is orientable or nonorientable, especially when we compare the graph to the graphs of the Möbius strip and torus in the previous examples.

A parametrization of the figure-8 immersion of the Klein bottle (see [20]) is $\mathbf{r}(s, t) = x(s, t)\mathbf{i} + y(s, t)\mathbf{j} + z(s, t)\mathbf{k}$, $-\pi \leq s \leq \pi$, $-\pi \leq t \leq \pi$, where

$$\begin{aligned} x &= \left[c + \cos\left(\frac{1}{2}s\right) \sin t - \sin\left(\frac{1}{2}s\right) \sin 2t \right] \cos s, \\ y &= \left[c + \cos\left(\frac{1}{2}s\right) \sin t - \sin\left(\frac{1}{2}s\right) \sin 2t \right] \sin s, \end{aligned} \quad (5.23)$$

and

$$z = \sin\left(\frac{1}{2}s\right) \sin t + \cos\left(\frac{1}{2}s\right) \sin 2t.$$

The plot in Figure 5.18(b) uses equation (5.23) if $c = 3$.

Using (5.21), let

$$\mathbf{n} = \frac{\mathbf{r}_s \times \mathbf{r}_t}{\|\mathbf{r}_s \times \mathbf{r}_t\|}.$$

Let C be the path given by

$$\mathbf{r}(t, t) = x(t, t)\mathbf{i} + y(t, t)\mathbf{j} + z(t, t)\mathbf{k}, \quad -\pi \leq t \leq \pi \quad (5.24)$$

that begins and ends at $\mathbf{r}(-\pi, -\pi) = \mathbf{r}(\pi, \pi) = \langle -3, 0, 0 \rangle$ and where the components are given by (5.23). The components of $\mathbf{r}$ and $\mathbf{n}$ are computed with Mathematica. The final calculations are quite lengthy, so we suppress the output of the last few by placing a semicolon (;) at the end of those commands.

rs = D[r[s, t], s]//Simplify

$$\begin{aligned} &\{-\frac{1}{2}\cos[s](\sin[\frac{s}{2}]\sin[t] + \cos[\frac{s}{2}]\sin[2t]) + \\ &\quad \sin[s](-3 - \cos[\frac{s}{2}]\sin[t] + \sin[\frac{s}{2}]\sin[2t]), \\ &\quad -\frac{1}{2}\sin[s](\sin[\frac{s}{2}]\sin[t] + \cos[\frac{s}{2}]\sin[2t]) + \\ &\quad \cos[s](3 + \cos[\frac{s}{2}]\sin[t] - \sin[\frac{s}{2}]\sin[2t]), \\ &\quad \frac{1}{2}(\cos[\frac{s}{2}] - 2\cos[t]\sin[\frac{s}{2}])\sin[t]\} \end{aligned}$$

rt = D[r[s, t], t]//Simplify

$$\begin{aligned} &\{\cos[s](\cos[\frac{s}{2}]\cos[t] - 2\cos[2t]\sin[\frac{s}{2}]), \\ &\quad (\cos[\frac{s}{2}]\cos[t] - 2\cos[2t]\sin[\frac{s}{2}])\sin[s], \\ &\quad 2\cos[\frac{s}{2}]\cos[2t] + \cos[t]\sin[\frac{s}{2}]\} \end{aligned}$$

rscrossrt = Cross[rs, rt];

normcross = Sqrt[rscrossrt . rscrossrt];

Clear[un]

un[s, t] = -rscrossrt / Sqrt[rscrossrt . rscrossrt];

At $t = -\pi$, $\mathbf{n}(-\pi, -\pi) = \left\langle \frac{1}{\sqrt{5}}, 0, \frac{2}{\sqrt{5}} \right\rangle$, whereas at $t = \pi$, $\mathbf{n}(\pi, \pi) = \left\langle -\frac{1}{\sqrt{5}}, 0, -\frac{2}{\sqrt{5}} \right\rangle$ so as $\mathbf{n}$ moves along C from $\mathbf{r}(-\pi, -\pi)$ to $\mathbf{r}(\pi, \pi)$, the orientation of $\mathbf{n}$ reverses. Several different views of the orientation reversing path on the Klein bottle shown in Figure 5.18(b) are shown in Figure 5.19.

l1 = Table[r[s, s], {s, -Pi, Pi, 2Pi/179}];

threep2 = Show[Graphics3D[{Thickness[.02],

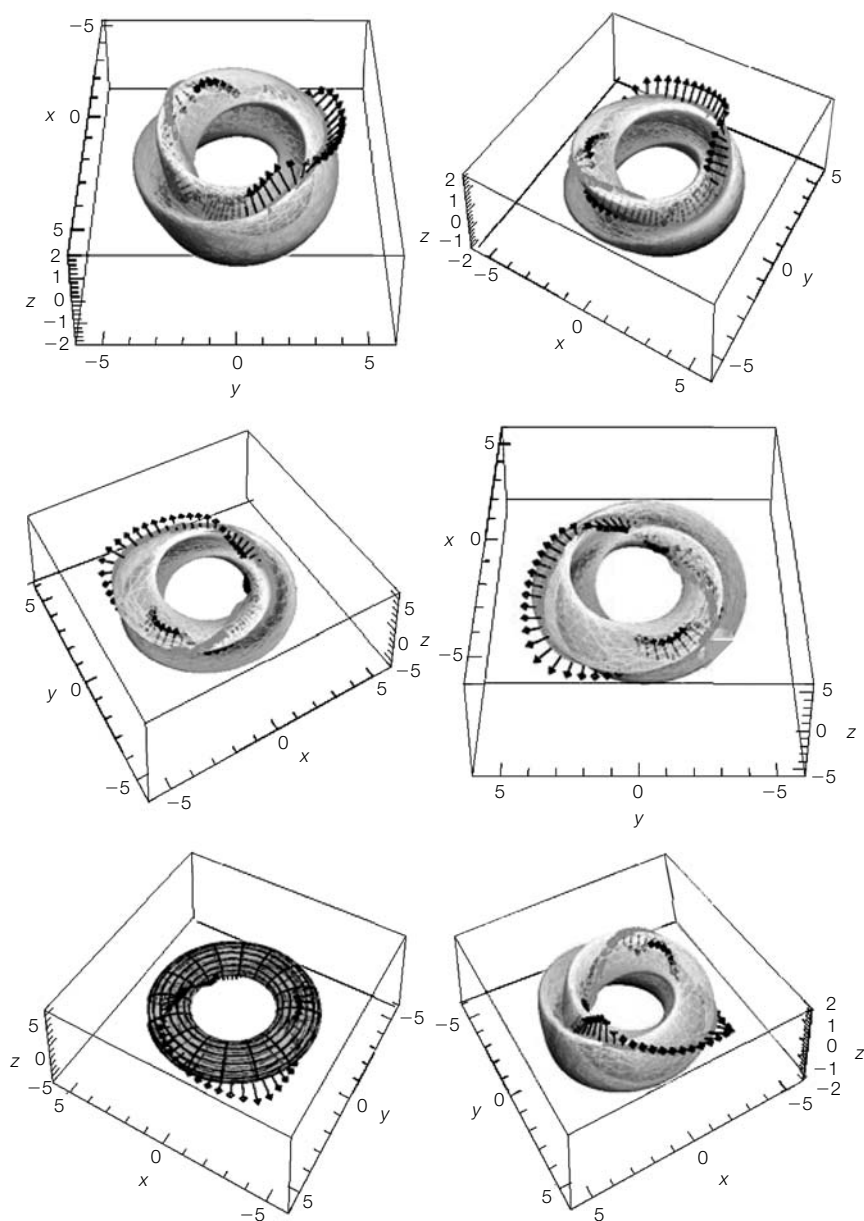
GrayLevel[.6], Line[l1]}], Axes->Automatic,

PlotRange->{{-4, 4}, {-4, 4}, {-4, 4}},

BoxRatios->{4, 4, 1}, AspectRatio->1];

<< "VectorFieldPlots";

vecs = Table[{r[s, s], un[s, s]}, {s, -pi, pi, 2pi/59}];

**FIGURE 5.19**

Different views of the figure-8 immersion of the Klein bottle with an orientation reversing path

```

pp2 = ListVectorFieldPlot3D[vecs, VectorHeads → True];
pp3 = Show[threedp2, pp2,
  AxesLabel → {"x", "y", "z"},
  Boxed → False, PlotRange → {{-5, 5},
    {-5, 5}, {-5, 5}}]
g1 = Show[kb2, threedp2, pp2, AspectRatio → 1,
  PlotRange → {{-6, 6}, {-6, 6}, {-2, 2}},
  BoxRatios → {4, 4, 1}, AxesLabel → {"x", "y", "z"},
  ViewPoint → {2.729, -0.000, 2.000}]
g2 = Show[kb2, threedp2, pp2,
  AspectRatio → 1,
  PlotRange → {{-6, 6}, {-6, 6}, {-2, 2}},
  BoxRatios → {4, 4, 1},
  AxesLabel → {"x", "y", "z"},
  ViewPoint → {1.365, -2.364, 2.000}]
g3 = Show[kb2, threedp2, pp2, AspectRatio → 1,
  PlotRange → {{-6, 6}, {-6, 6}, {-6, 6}},
  BoxRatios → {4, 4, 1}, AxesLabel → {"x", "y", "z"},
  ViewPoint → {-1.365, -2.364, 2.000}]
g4 = Show[kb2, threedp2, pp2, AspectRatio → 1,
  PlotRange → {{-6, 6}, {-6, 6}, {-6, 6}},
  BoxRatios → {4, 4, 1}, AxesLabel → {"x", "y", "z"},
  ViewPoint → {-2.729, 0.000, 2.000}]
g5 = Show[threedp1t, pp2, AspectRatio → 1,
  PlotRange → {{-6, 6}, {-6, 6}, {-6, 6}},
  BoxRatios → {4, 4, 1}, AxesLabel → {"x", "y", "z"},
  ViewPoint → {-1.365, 2.364, 2.000}]
g6 = Show[kb2, pp3, AspectRatio → 1,
  PlotRange → {{-6, 6}, {-6, 6}, {-2, 2}},
  BoxRatios → {4, 4, 1}, AxesLabel → {"x", "y", "z"},
  ViewPoint → {1.365, 2.364, 2.000}]
Show[GraphicsGrid[{{g1, g2}, {g3, g4}, {g5, g6}}]]

```

These concepts are presented *beautifully* and extensively for the Mathematica user in *Modern Differential Geometry of Curves and Surfaces with Mathematica*, third edition, by Alfred Gray, Elsa Abbena, and Simon Salamon. Our treatment just touches on a few of the topics discussed by Gray et al and updates some of their wonderful and elegant work to Mathematica 6.

C is an orientation reversing path and we can conclude that the Klein bottle is not orientable.

5.5.5 More on Tangents, Normals, and Curvature in $\mathcal{R}^3$

Previously, we discussed the unit tangent and normal vectors and curvature for a vector-valued function $\gamma : (a, b) \rightarrow \mathcal{R}^2$. These concepts can be extended to curves and surfaces in space.

For $\gamma : (a, b) \rightarrow \mathcal{R}^3$, the **Frenet frame field** is the ordered triple $\{\mathbf{T}, \mathbf{N}, \mathbf{B}\}$, where $\mathbf{T}$ is the **unit tangent vector field**, $\mathbf{N}$ is the **unit**

For many good reasons, sometimes the “Frenet formulas” are also called the “Frenet-Serret formulas.”

normal vector field, and **B** is the **unit binormal vector field**. Each of these vectors has norm 1 and each is orthogonal to the other (the dot product of one with another is 0) and the **Frenet formulas** are satisfied: $\mathbf{T}' = \kappa \mathbf{N}$, $\mathbf{N}' = -\kappa \mathbf{T} + \tau \mathbf{B}$, $\mathbf{B}' = -\tau \mathbf{N}$. τ is the **torsion** of the curve γ ; κ is the curvature. For the curve $\gamma : (a, b) \rightarrow \mathcal{R}^3$, formulas for these quantities are given by

$$\begin{aligned} \mathbf{T} &= \frac{\gamma'}{\|\gamma'\|}, & \mathbf{N} &= \mathbf{B} \times \mathbf{T}, & \mathbf{B} &= \frac{\gamma' \times \gamma''}{\|\gamma' \times \gamma''\|}, \\ \kappa &= \frac{\|\gamma' \times \gamma''\|}{\|\gamma'\|^3}, & \tau &= \frac{\gamma' \times \gamma'' \cdot \gamma'''}{\|\gamma' \times \gamma''\|^2}. \end{aligned} \quad (5.25)$$

We adjust Gray’s routines slightly for Mathematica 6. Here is the unit tangent vector:

```
tangent[γ _][t _]:=D[γ[tt], tt]/FullSimplify[Norm[D[γ[tt], tt]],
Assumptions → tt ∈ Reals]/.tt → t
```

Similarly, the binormal is defined with

```
binormal[γ _][t _]:=FullSimplify[
Cross[D[γ[tt], tt], D[γ[tt], {tt, 2}]]/
FullSimplify[Norm[Cross[D[γ[tt], tt], D[γ[tt], {tt, 2}]]],
Assumptions → tt ∈ Reals]/.tt → t
```

so the unit normal is defined with

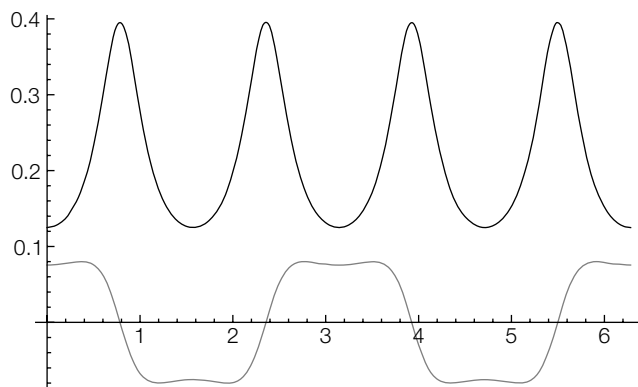
```
normal[γ _][t _]:=Cross[binormal[γ][t], tangent[γ][t];
```

Notice how we use Assumptions to instruct Mathematica to assume that the domain of γ consists of real numbers. In the same manner, we define the curvature and torsion.

```
curve2[γ _][t _]:=Simplify[Norm[Cross[D[γ[tt], tt],
D[γ[tt], {tt, 2}]]]/
Norm[D[γ[tt], tt]]^3,
Assumptions → tt ∈ Reals]/.tt → t;

torsion2[γ _][t _]:=Simplify[Cross[D[γ[tt], tt],
D[γ[tt], {tt, 2}]] . D[γ[tt], {tt, 3}]/
Norm[Cross[D[γ[tt], tt], D[γ[tt], {tt, 2}]]]^2,
Assumptions → tt ∈ Reals]/.tt → t;
```

In even the simplest situations, these calculations are quite complicated. Graphically seeing the results may be more meaningful than the explicit formulas.

**FIGURE 5.20**

The curvature and torsion for a spherical spiral

Example 5.5.11 Consider the spherical spiral given by $\gamma(t) = \langle 8 \cos 3t \cos 2t, 8 \sin 3t \cos 2t, 8 \sin 2t \rangle$. The curvature and torsion for the curve are graphed with Plot and shown in Figure 5.20.

```
<< VectorFieldPlots`;  
γ[t_] = {8Cos[3t]Cos[2t], 8Sin[3t]Cos[2t], 8Sin[2t]}  
{8Cos[2t]Cos[3t], 8Cos[2t]Sin[3t], 8Sin[2t]}  
Plot[Tooltip[{curve2[γ][t], torsion2[γ][t]}], {t, 0, 2Pi},  
PlotStyle -> {Black, Gray}]
```

We now compute $\mathbf{T}$, $\mathbf{B}$, and $\mathbf{N}$. For length considerations, we display an abbreviated portion of $\mathbf{B}$ with Short.

```
tangent[γ][t]  
binormal[γ][t]  
normal[γ][t]/Short
```

$$\left\{ \begin{array}{l} \frac{-16\cos[3t]\sin[2t] - 24\cos[2t]\sin[3t]}{4\sqrt{34 + 18\cos[4t]}}, \\ \frac{24\cos[2t]\cos[3t] - 16\sin[2t]\sin[3t]}{4\sqrt{34 + 18\cos[4t]}}, \\ \frac{4\cos[2t]}{\sqrt{34 + 18\cos[4t]}} \end{array} \right\}$$

$$\left\{ \begin{array}{l} \frac{2(-3\sin[t] + 34\sin[3t] + 15\sin[7t])}{\sqrt{21886 + 12456\cos[4t] + 810\cos[8t]}}, \\ -\frac{2(3\cos[t] + 34\cos[3t] + 15\cos[7t])}{\sqrt{21886 + 12456\cos[4t] + 810\cos[8t]}}, \\ \frac{6(21 + 5\cos[4t])}{\sqrt{21886 + 12456\cos[4t] + 810\cos[8t]}} \end{array} \right\}$$

$$\left\{ \langle\langle 7 \rangle\rangle + \frac{\langle\langle 1 \rangle\rangle}{\langle\langle 1 \rangle\rangle} + \frac{120\langle\langle 2 \rangle\rangle \sin[3t]}{\sqrt{34 + 18\cos[\langle\langle 1 \rangle\rangle]}\sqrt{1}}, \langle\langle 1 \rangle\rangle, \langle\langle 1 \rangle\rangle \right\}$$

It is difficult to see how these complicated formulas relate to this spherical spiral. To help us understand what they mean, we first plot the spiral with `ParametricPlot3D`. See Figure 5.21(a).

```
p1 = ParametricPlot3D[ $\gamma[t]$ , {t, 0, 2Pi},
  PlotRange → {{-8.5, 8.5}, {-8.5, 8.5}, {-8.5, 8.5}},
  PlotStyle → {{Gray, Thick}}]
```

Next, we use `Table` to compute lists of two ordered triples. For each list, the first ordered triple consists of $\gamma(t)$ and the second the value of $\mathbf{T}(\gamma(t))$ ($\mathbf{B}(\gamma(t))$, $\mathbf{N}(\gamma(t))$). These ordered triples that correspond to vectors are plotted with `ListVectorFieldPlot3D`, which is contained in the **VectorFieldPlots** package, in Figure 5.21(b).

```
ts = Table[{ $\gamma[t]$ , tangent[ $\gamma[t]$ ]/N, {t, 0, 2Pi, 2Pi/99}};
bs = Table[{ $\gamma[t]$ , binormal[ $\gamma[t]$ ]/N, {t, 0, 2Pi, 2Pi/99}};
ns = Table[{ $\gamma[t]$ , normal[ $\gamma[t]$ ]/N, {t, 0, 2Pi, 2Pi/99}};

ysplot = ListVectorFieldPlot3D[ts, VectorHeads → True];
bsplot = ListVectorFieldPlot3D[bs, VectorHeads → True];
nsplot = ListVectorFieldPlot3D[ns, VectorHeads → True];
p2 = Show[ysplot, bsplot, nsplot]
```

For a good view of `p1` and `p2`, display them together with `Show`. See Figure 5.21(c).

```
Show[p1, p2]Show[GraphicsRow[{p1, p2, Show[p1, p2]}]]
```

The previous example illustrates that capturing the depth of three-dimensional curves by projections into two dimensions can be difficult. Sometimes taking advantage of three-dimensional surface plots can help.

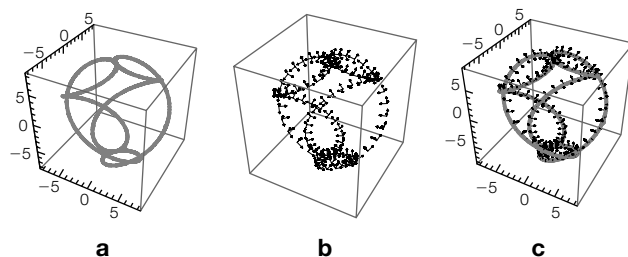


FIGURE 5.21

(a) The spherical spiral. (b) Various $\mathbf{T}$, $\mathbf{N}$, and $\mathbf{B}$ for the spherical spiral. (c) The spherical spiral with various $\mathbf{T}$, $\mathbf{N}$, and $\mathbf{B}$ shown together

For a basic space curve, tubecurve places a “tube” of radius r around the space curve.

```
Clear[tubecurve]
tubecurve[γ_][r_][t_ , θ_] = γ[t] +
  r(Cos[θ]normal[γ][t] + Sin[θ]binormal[γ][t])
r(Cos[θ]  $\frac{\gamma'[t] \times \gamma''[t]}{\text{Norm}[\gamma'[t] \times \gamma''[t]]} \times \frac{\gamma'[t]}{\text{Norm}[\gamma'[t]]} + \frac{\gamma'[t] \times \gamma''[t] \text{Sin}[\theta]}{\text{Norm}[\gamma'[t] \times \gamma''[t]]}$ ) + γ[t]
```

The results displayed in the text are in black-and-white and do not reflect the stunning color images generated by these commands.

To illustrate the utility, we redefine torusknot that was presented in Chapter 2.

```
torusknot[a_ , b_ , c_][p_ , q_][t_] :=
  {(a + b Cos[q t])Cos[p t], (a + b Cos[q t])Sin[p t],
   c Sin[q t]}
```

Example 5.5.12 For the knot `torusknot[8,3,5][2,5]` we plot the curvature and torsion with `Plot`.

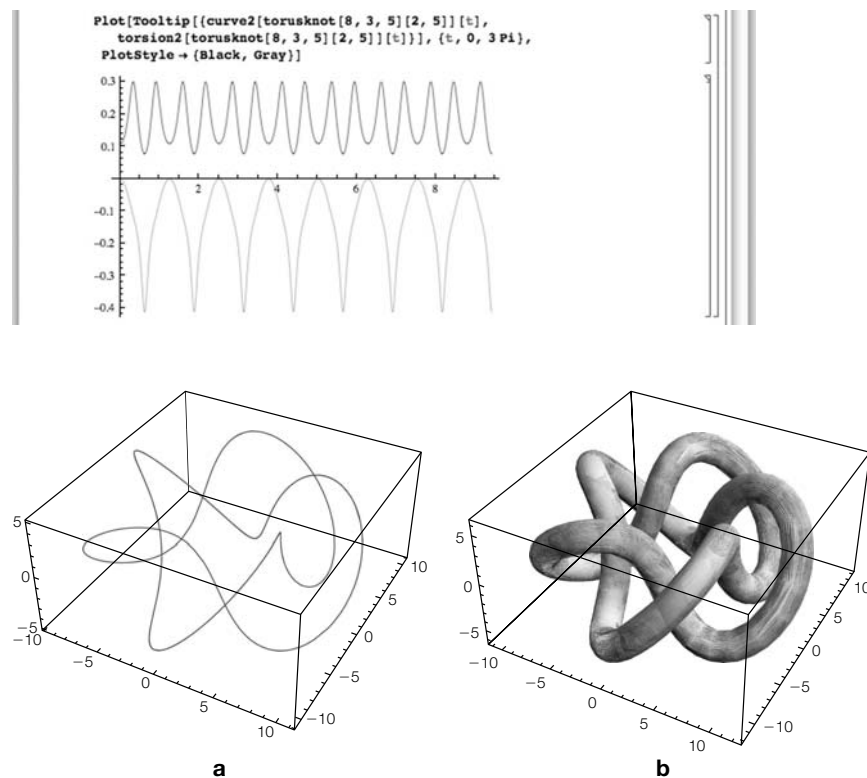


FIGURE 5.22

(a) A basic plot of a curve in 3-space. (b) Placing a “tube” around the curve

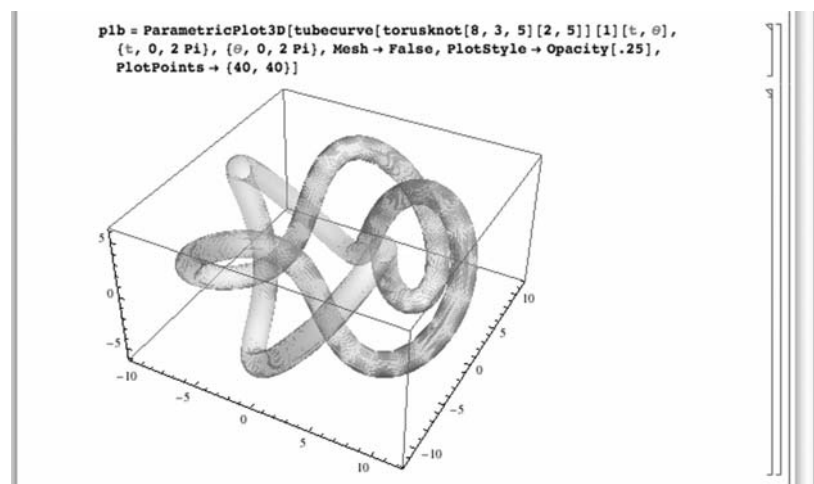
We generate a basic plot of this torus knot in 3-space with `ParametricPlot3D`. See Figure 5.22(a).

`ParametricPlot3D[torusknot[8, 3, 5][2, 5][t], {t, 0, 3Pi}]`

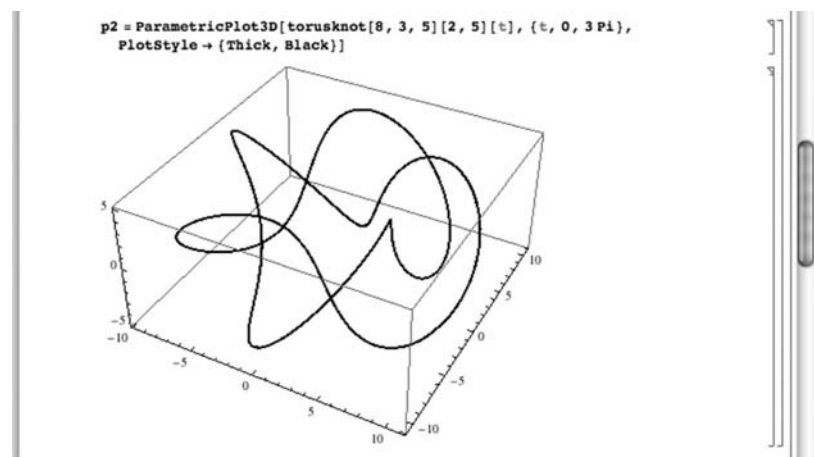
Using `tubeplot`, we place a “tube” around the knot. See Figure 5.22(b).

**`p1 = ParametricPlot3D[tubecurve[torusknot[8, 3, 5][2, 5][1][t, θ],
{t, 0, 2Pi}, { θ , 0, 2Pi}], Mesh $\rightarrow$ False, PlotStyle $\rightarrow$ Opacity[.5],
PlotPoints $\rightarrow$ {40, 40}]`**

A more interesting graphic is obtained by placing a transparent tube around the curve



and then creating a thicker version of the curve.



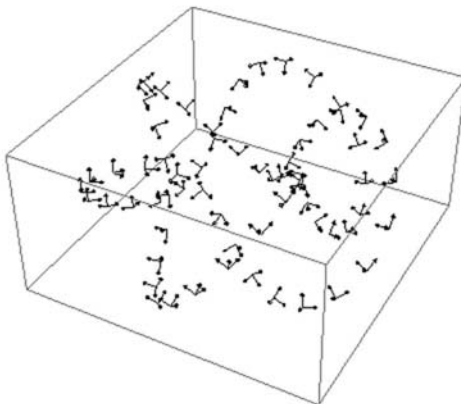
As before, we use `tangent`, `normal`, and `binormal` to create a vector field on the curve.

```

ts = Table[{torusknot[8, 3, 5][2, 5][t],
  tangent[torusknot[8, 3, 5][2, 5][t]] // N, {t, 0, 3 Pi, 3 Pi / 99}};
bs = Table[{torusknot[8, 3, 5][2, 5][t],
  binormal[torusknot[8, 3, 5][2, 5][t]] // N, {t, 0, 3 Pi, 3 Pi / 99}};
ns = Table[{torusknot[8, 3, 5][2, 5][t],
  normal[torusknot[8, 3, 5][2, 5][t]] // N, {t, 0, 3 Pi, 3 Pi / 99}};

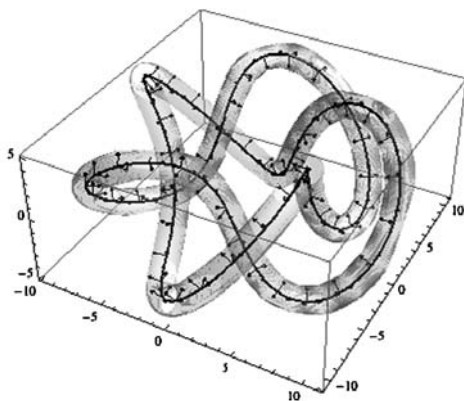
ysplot = ListVectorFieldPlot3D[ts, VectorHeads -> True];
bsplot = ListVectorFieldPlot3D[bs, VectorHeads -> True];
nsplot = ListVectorFieldPlot3D[ns, VectorHeads -> True];
p3 = Show[ysplot, bsplot, nsplot]

```



A striking graph is generated by showing the three graphs together.

```
Show[p2, p1b, p3]
```



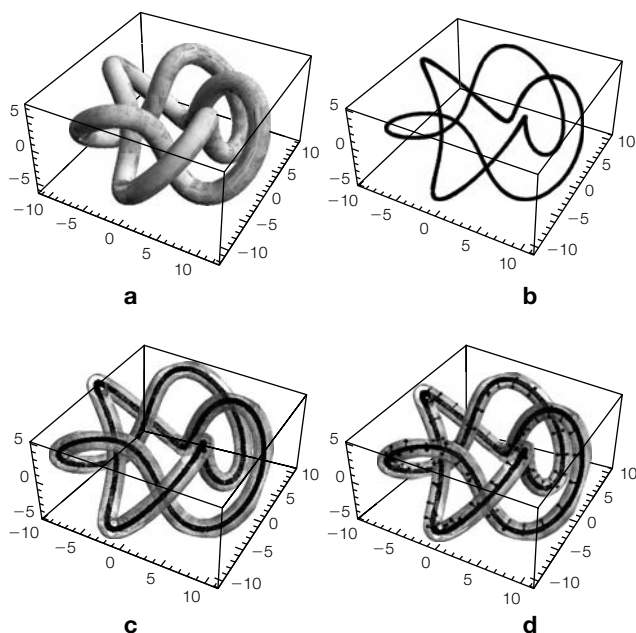


FIGURE 5.23

(a) A “tubed” knot. (b) A thick knot. (c) A knot within a tube around it. (d) A knot within a tube illustrating the Frenet field

Alternatively, display the results as an array with `GraphicsGrid`. See Figure 5.23.

```
Show[GraphicsGrid[{{p1, p2}, {Show[p2, p1b], Show[p2, p1b, p3]]}}]
```

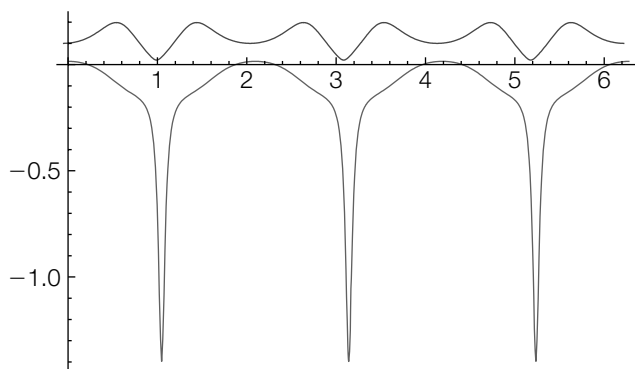
Example 5.5.13 The **Trefoil knot** is the special case of `torusknot[8,3,5][2,3][t]`. We use `Plot` to graph its curvature and torsion in Figure 5.24. Because we have used `Tooltip`, you can identify each plot by moving the cursor over the curve in Figure 5.24.

```
Plot[Tooltip[{curve2[torusknot[8, 3, 5][2, 3][t],
torsion2[torusknot[8, 3, 5][2, 3][t]], {t, 0, 2Pi}], PlotRange → All]
```

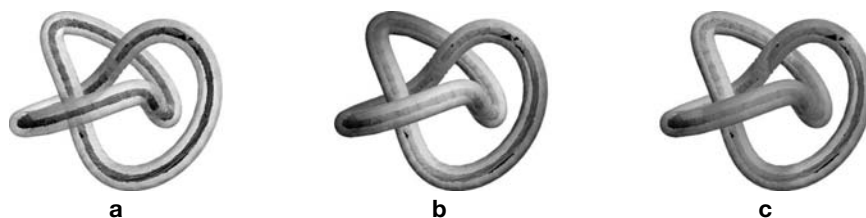
Next, we generate a thickened version of the Trefoil knot.

```
p1 = ParametricPlot3D[torusknot[8, 3, 5][2, 3][t], {t, 0, 2Pi},
PlotStyle → {Black, Thick}]
```

Three different tube plots of the Trefoil knot are generated. In `p2`, the result is a basic plot. In `p2b`, the plot is shaded according to the `Rainbow` color gradient.

**FIGURE 5.24**

The curvature and torsion for the Trefoil knot

**FIGURE 5.25**

(a) The Trefoil knot with a tube around it. (b) Changing the color of the tube. (c) Coloring the knot according to its curvature

In p2c, the plot is shaded according to the knot's curvature. The knot together with the three surfaces are shown in Figure 5.25.

```
p2 = ParametricPlot3D[tubecurve[torusknot[8, 3, 5][2, 3]][1.3][t,  $\theta$ ],
  {t, 0, 2Pi}, { $\theta$ , 0, 2Pi}, Mesh → False, PlotStyle → Opacity[.5],
  PlotPoints → {40, 40}]
p2b = ParametricPlot3D[tubecurve[torusknot[8, 3, 5][2, 3]][1.3][t,  $\theta$ ],
  {t, 0, 2Pi}, { $\theta$ , 0, 2Pi}, Mesh → False, PlotStyle → Opacity[.5],
  PlotPoints → {40, 40}, ColorFunction → ColorData["Rainbow"]]
p2c = ParametricPlot3D[tubecurve[torusknot[8, 3, 5][2, 3]][1.3][t,  $\theta$ ],
  {t, 0, 2Pi}, { $\theta$ , 0, 2Pi}, Mesh → False, PlotStyle → Opacity[.5],
  PlotPoints → {40, 40}, ColorFunction →
    (ColorData["BrightBands"])[curve2[torusknot[8, 3, 5][2, 3]][#1]]&)]
ba1 = Show[p2, p1, Boxed → False, Axes → None]
ba2 = Show[p2b, p1, Boxed → False, Axes → None]
ba3 = Show[p2c, p1, Boxed → False, Axes → None]
Show[GraphicsRow[{ba1, ba2, ba3}]]
```

For surfaces in $\mathcal{R}^3$, extending and stating these definitions precisely becomes even more complicated. First, define the **vector triple product** (xyz) , where $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, $\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$, and $\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$, by $(xyz) =$

$$\begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{vmatrix}.$$

We assume that $\gamma = \gamma(u, v)$ is a vector-valued function with domain contained in a “nice” region $U \subset \mathcal{R}^2$ and range in $\mathcal{R}^3$. The **Gaussian curvature**, $\mathcal{K}$, and the **mean curvature**, $\mathcal{H}$, under reasonable conditions, are given by the formulas

$$\mathcal{K} = \frac{(\gamma_{uu}\gamma_u\gamma_v)(\gamma_{vv}\gamma_u\gamma_v) - (\gamma_{uv}\gamma_u\gamma_v)^2}{\left(\|\gamma_u\|^2\|\gamma_v\|^2 - (\gamma_u \cdot \gamma_v)^2\right)^2}$$

and (5.26)

$$\mathcal{H} = \frac{(\gamma_{uu}\gamma_u\gamma_v)\|\gamma_v\|^2 - 2(\gamma_{uv}\gamma_u\gamma_v)(\gamma_u \cdot \gamma_v) + (\gamma_{vv}\gamma_u\gamma_v)\|\gamma_u\|^2}{2\left(\|\gamma_u\|^2\|\gamma_v\|^2 - (\gamma_u \cdot \gamma_v)^2\right)^{3/2}}.$$

For the parametrically defined surface $\gamma = \gamma(u, v)$, the **unit normal field**, $\mathbf{U}$, is $\mathbf{U} = \frac{\gamma_u \times \gamma_v}{\|\gamma_u \times \gamma_v\|}$. Observe that the expressions that result from explicitly computing $\mathbf{U}$, $\mathcal{K}$, and $\mathcal{H}$ are almost always so complicated that they are impossible to understand.

After defining `vtp` to return the vector triple product of three vectors, we define `gaussanc` and `meanc` to compute $\mathcal{K}$ and $\mathcal{H}$ for a parametrically defined surface $\gamma(u, v) = \langle x(u, v), y(u, v), z(u, v) \rangle$.

```

vtp[x_, y_, z_] := Det[{{x[[1]], x[[2]], x[[3]]},
{y[[1]], y[[2]], y[[3]]}, {z[[1]], z[[2]], z[[3]]}}]

gaussanc[γ_][u_, v_] :=
Module[{lu, lv, vtp},
vtp[x_, y_, z_] := Det[{{x[[1]], x[[2]], x[[3]]},
{y[[1]], y[[2]], y[[3]]}, {z[[1]], z[[2]], z[[3]]}}];
(vtp[D[γ[lu, lv], {lu, 2}], D[γ[lu, lv], lv], D[γ[lu, lv], lv]]
vtp[D[γ[lu, lv], {lv, 2}], D[γ[lu, lv], lu], D[γ[lu, lv], lv]] -
vtp[D[γ[lu, lv], lu, lv], D[γ[lu, lv], lu], D[γ[lu, lv], lv]] ^ 2) /
(Norm[D[γ[lu, lv], lu]] ^ 2 Norm[D[γ[lu, lv], lv]] ^ 2 -
(D[γ[lu, lv], lu] . D[γ[lu, lv], lv]) ^ 2) ^ 2 /.
{lu -> u, lv -> v} // PowerExpand // Simplify
]

```

```

meanc[γ _][u _, v _]:=
Module[{lu, lv, vtp},
vtp[x _, y _, z _]:=Det[{{x[[1]], x[[2]], x[[3]]},
{y[[1]], y[[2]], y[[3]]}, {z[[1]], z[[2]], z[[3]]}}];
(vtp[D[γ[lu, lv], {lu, 2}], D[γ[lu, lv], lu], D[γ[lu, lv], lv]]
Norm[D[γ[lu, lv], lv]] ^ 2 -
2vtp[D[γ[lu, lv], lu, lv], D[γ[lu, lv], lu], D[γ[lu, lv], lv]]
(D[γ[lu, lv], lu] . D[γ[lu, lv], lv]) +
vtp[D[γ[lu, lv], {lv, 2}], D[γ[lu, lv], lu], D[γ[lu, lv], lv]]
Norm[D[γ[lu, lv], lu]] ^ 2) /
(2(Norm[D[γ[lu, lv], lu]] ^ 2 Norm[D[γ[lu, lv], lv]] ^ 2 -
(D[γ[lu, lv], lu] . D[γ[lu, lv], lv]) ^ 2) ^ (3/2))/
{lu → u, lv → v} // PowerExpand // Simplify
]

```

Example 5.5.14 We illustrate the commands with the torus, first discussed in Chapter 2, and ParametricPlot3D. For convenience, we redefine torus.

```

torus[a _, b _, c _][p _, q _][u _, v _]:={{(a + bCos[u])
Cos[v], (a + b Cos[u])Sin[v], c Sin[u]}

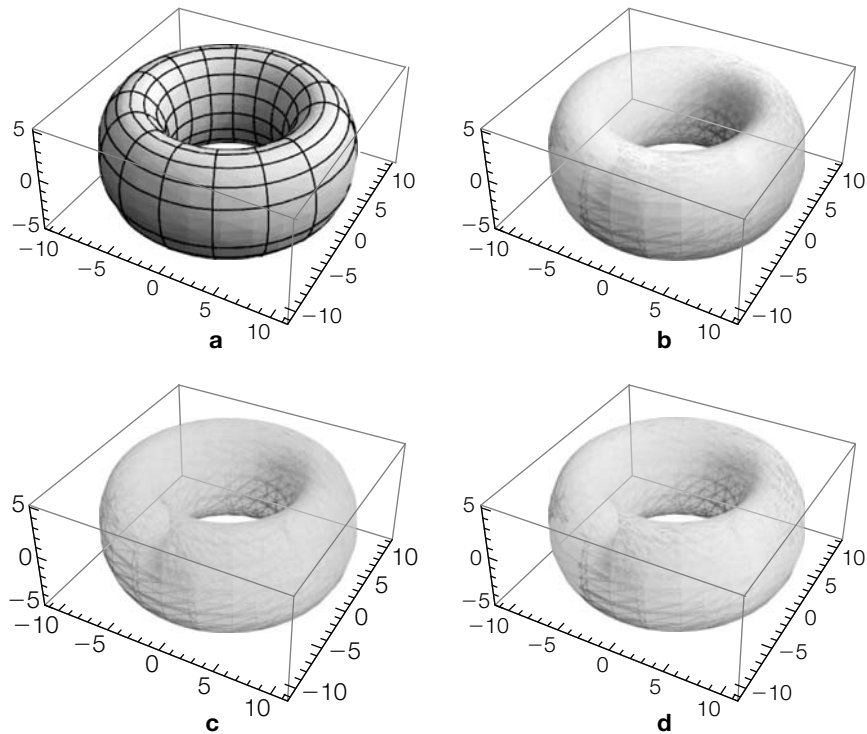
```

In pp1, we generate a basic plot of the torus. The shading is changed in pp2. In pp3 the surface is shaded according to its Gaussian curvature, whereas in pp4 the surface is shaded according to its mean curvature. All four plots are shown together in Figure 5.26.

```

pp1 = ParametricPlot3D[Evaluate[torus[8, 3, 5][2, 5][u, v]], {u, 0, 2Pi},
{v, 0, 2Pi}, PlotPoints → 60]
pp2 = ParametricPlot3D[torus[8, 3, 5][2, 5][u, v],
{u, 0, 2Pi}, {v, 0, 2Pi}, Mesh → False, PlotStyle → Opacity[.75],
PlotPoints → {25, 25}, ColorFunction →
ColorData["MintColors"]]
pp3 = ParametricPlot3D[torus[8, 3, 5][2, 5][u, v],
{u, 0, 2Pi}, {v, 0, 2Pi}, Mesh → False, PlotStyle → Opacity[.5],
PlotPoints → {25, 25}, ColorFunction →
(ColorData["MintColors"])[gaussanc[torus[8, 3, 5][2, 5]]
[#1, #2]//N//Chop]&]]
pp4 = ParametricPlot3D[torus[8, 3, 5][2, 5][u, v],
{u, 0, 2Pi}, {v, 0, 2Pi}, Mesh → False, PlotStyle → Opacity[.5],
PlotPoints → {25, 25}, ColorFunction →
(ColorData["MintColors"])[meanc[torus[8, 3, 5][2, 5]]
[#1, #2]//N//Chop]&]]
Show[GraphicsGrid[{{pp1, pp2}, {pp3, pp4}}]]

```

**FIGURE 5.26**

(a) A basic torus. (b) Changing the coloring of the torus. (c) Shading according to Gaussian curvature. (d) Shading according to mean curvature

5.6 MATRICES AND GRAPHICS

Mathematica contains several functions that allow you to represent matrices graphically. These commands are analogous to the corresponding ones for dealing with lists (such as `ListPlot`) or functions (such as `Plot`, `Plot3D`, and `ContourPlot`).

1. `MatrixPlot[A]` generates a grid with the same dimensions as **A**. The cells are shaded according to the entries of **A**. The default is in color.
2. `ArrayPlot[A]` generates a grid with the same dimensions as **A**. The cells are shaded according to the entries of **A**. The default is in black and white.
3. `ListContourPlot[A]` generates a contour plot using the entries of **A** as the height values.
4. `ReliefPlot[A]` generates a relief plot using the entries of **A** as the height values.

Because the figures in the text are in black and white, refer to the CD that accompanies the text to see the images in color.

Observe that `ArrayPlot` and `MatrixPlot` are virtually interchangeable. However, the entries of `ArrayPlot` need not be numbers. If Mathematica cannot determine how to shade a cell, the default is to shade it in a dark maroon color. Although these functions generate graphics that depend on the entries of the matrix, loosely speaking we will use phrases such as “we use `MatrixPlot` to plot *A*” and “we use `ArrayPlot` to graph *A*” to describe the graphic that results from applying one of these functions to an array.

For example, consider the arrays $A = \begin{pmatrix} 1 & 0 & .3 \\ .4 & .5 & .1 \\ .2 & .3 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 & 0 \\ .1 & .2 & .3 \end{pmatrix}$, and $C = \begin{pmatrix} (1 & 0 & 0) & (0 & 1 & 0) \\ (.3 & .4 & .5) & (.1 & .2 & .3) \end{pmatrix}$.

In the first command, Mathematica shades all the cells according to its `GrayLevel` value. However, in the second and third commands, Mathematica cannot shade the cells in the second row and all the cells, respectively, because ordered triples cannot be evaluated by `GrayLevel`. However, `RGBColor` evaluates ordered triples so Mathematica shades the cells in Figure 5.27(c) according to their `RGBColor` value.

```
ap1 = ArrayPlot[{{1, 0, .3}, {.4, .5, .1}, {.2, .3, 0}}];
ap2 = ArrayPlot[{{1, 0}, {.3, .4, .5}, {.1, .2, .3}}];
ap3 = ArrayPlot[{{{1, 0, 0}, {0, 1, 0}}, {{.3, .4, .5},
    {.1, .2, .3}}];
ap4 = ArrayPlot[{{{1, 0, 0}, {0, 1, 0}}, {{.3, .4, .5},
    {.1, .2, .3}}],
    ColorFunction -> RGBColor];
Show[GraphicsRow[{ap1, ap2, ap3, ap4}]]
```

`MatrixPlot` is unable to graphically represent *B* or *C*. However, coloring is automatic with `MatrixPlot`. See Figure 5.28.

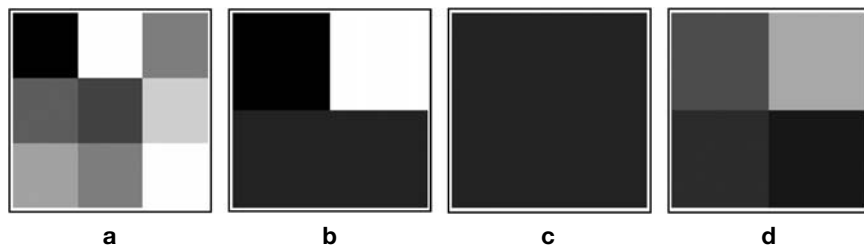
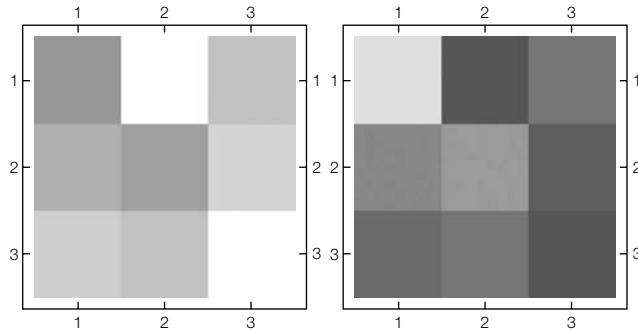
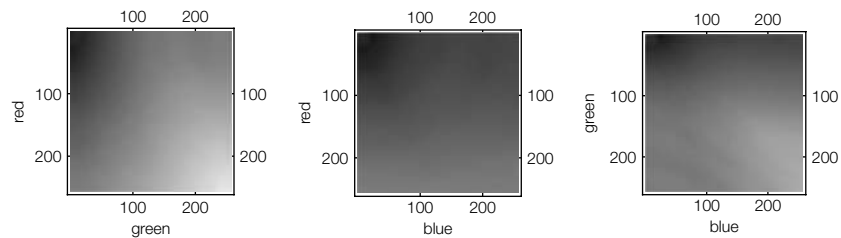


FIGURE 5.27

(a) Mathematica shades all cells according to their heights. (b) Mathematica does not know how to shade the cells in the second row. (c) Mathematica cannot shade any of the cells. (d) Mathematica shades all four cells using `RGBColor`

**FIGURE 5.28**

By default, `MatrixPlot` uses a color scheme. Use `ColorFunction` to change the colors

**FIGURE 5.29**

A comparison of how red, green, and blue affect `RGBColor[r,g,b]`

```
mp1 = MatrixPlot[{{1, 0, .3}, {4, .5, .1}, {2, .3, 0}}];
mp2 = MatrixPlot[{{1, 0, .3}, {4, .5, .1}, {2, .3, 0}},
  ColorFunction -> "PlumColors"];
Show[GraphicsRow[{mp1, mp2}]]
```

If you need to adjust the color of a graphic, usually you can use the **ColorSchemes** palette to select an appropriate gradient or color function. In other situations, you might wish to create your own using `Blend`. To use `Blend`, you might need to know how various `RGBColors` or `CMYKColors` vary as the variables affecting the color change.

`ArrayPlot` can help us see the variability in the colors. With the following, we see how `RGBColor[r,g,b]` affects color for $b = 0$, $g = 0$, and then $r = 0$. The results are shown together in Figure 5.29. The figure can help us select appropriate values to generate our own color blending function using `Blend` rather than relying on Mathematica's built-in color schemes and gradients.

```
t1 = Table[{r, g, 0}/N, {r, 0, 255}, {g, 0, 255}];
redgreen = ArrayPlot[t1, Axes -> Automatic, AxesOrigin -> {0, 0},
```

In these calculations, `t1` is a 256×256 array for which each entry is an ordered triple. In the first `t1`, the ordered triple has the form $(r, g, 0)$, in the second the form $(r, 0, b)$, and so on.

```

FrameTicks → Automatic, FrameLabel → {red, green},
LabelStyle → Medium, ColorFunction → RGBColor];
t2 = Table[{r, 0, b}/N, {r, 0, 255}, {b, 0, 255}];
redblue = ArrayPlot[t2, Axes → Automatic, AxesOrigin → {0, 0},
FrameTicks → Automatic, FrameLabel → {red, blue},
LabelStyle → Medium, ColorFunction → RGBColor];
t2 = Table[{0, g, b}/N, {g, 0, 255}, {b, 0, 255}];
greenblue = ArrayPlot[t2, Axes → Automatic, AxesOrigin → {0, 0},
FrameTicks → Automatic, FrameLabel → {green, blue},
LabelStyle → Medium, ColorFunction → RGBColor];
s1 = Show[GraphicsRow[{redgreen, redblue, greenblue}]]

```

We modify the calculation slightly to see how CMYKColor varies as we adjust two parameters. Keep in mind that each $t2$ is a 256×256 array. Each entry of $t2$ is an ordered quadruple, which is illustrated in the first calculation, in which we use Part to take the fifth element of the eighth part of $t2$. (See Figure 5.30.)

```

t2 = Table[{c, m, 0, 0}/N, {c, 0, 255}, {m, 0, 255}];
t2[[8, 5]]
{7., 4., 0., 0.}
cmplot = ArrayPlot[t2, Axes → Automatic, AxesOrigin → {0, 0},
FrameTicks → Automatic, FrameLabel → {c, m},
LabelStyle → Medium, ColorFunction → CMYKColor];
t2 = Table[{c, 0, y, 0}/N, {c, 0, 255}, {y, 0, 255}];

```

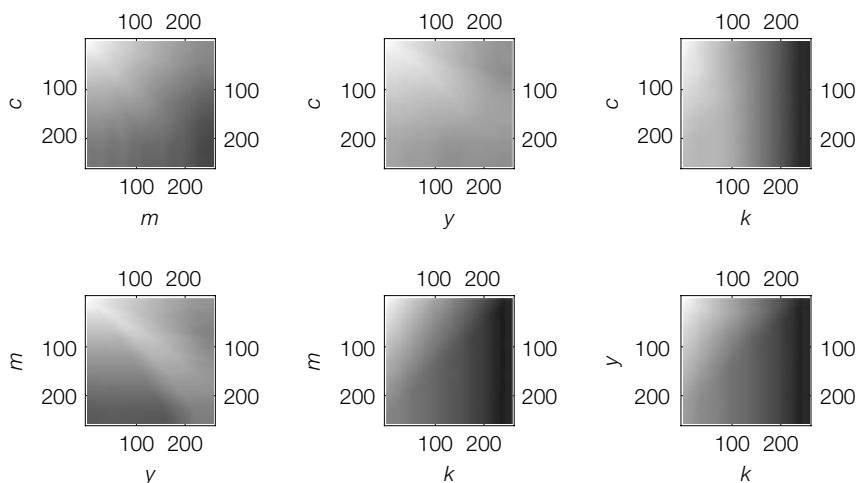


FIGURE 5.30

A comparison of how c , m , y , and k affect $\text{CMYKColor}[c,m,y,k]$

```

cyp1ot = ArrayPlot[t2, Axes → Automatic, AxesOrigin → {0, 0},
  FrameTicks → Automatic, FrameLabel → {c, y},
  LabelStyle → Medium, ColorFunction → CMYKColor];
t2 = Table[{c, 0, 0, k}/N, {c, 0, 255}, {k, 0, 255}];
ckp1ot = ArrayPlot[t2, Axes → Automatic, AxesOrigin → {0, 0},
  FrameTicks → Automatic, FrameLabel → {c, k},
  LabelStyle → Medium, ColorFunction → CMYKColor];
t2 = Table[{0, m, y, 0}/N, {m, 0, 255}, {y, 0, 255}];
myp1ot = ArrayPlot[t2, Axes → Automatic, AxesOrigin → {0, 0},
  FrameTicks → Automatic, FrameLabel → {m, y},
  LabelStyle → Medium, ColorFunction → CMYKColor];
t2 = Table[{0, m, 0, k}/N, {m, 0, 255}, {k, 0, 255}];
mkp1ot = ArrayPlot[t2, Axes → Automatic, AxesOrigin → {0, 0},
  FrameTicks → Automatic, FrameLabel → {m, k},
  LabelStyle → Medium, ColorFunction → CMYKColor];
t2 = Table[{0, 0, y, k}/N, {y, 0, 255}, {k, 0, 255}];
ykp1ot = ArrayPlot[t2, Axes → Automatic, AxesOrigin → {0, 0},
  FrameTicks → Automatic, FrameLabel → {y, k},
  LabelStyle → Medium, ColorFunction → CMYKColor];
Show[GraphicsGrid[{{cmplot, cyp1ot, ckp1ot}, {myp1ot, mkp1ot, ykp1ot}}]]

```

You can load files into Mathematica with `Import`. Generally, the underlying structure of the loaded file is relatively easy to understand. Be careful when you import data into Mathematica. We recommend that you use `ExampleData` to investigate your routines before finalizing them. Although importing external files into Mathematica is easy, understanding the underlying structure of the imported data may take some time but may be necessary to produce the results you desire.

We illustrate a few of the subtle differences that can be encountered with several color and black-and-white gifs and jpegs.

Using `Import`, we import a graphic of the primary author into Mathematica. The result is shown in Figure 5.31(a).

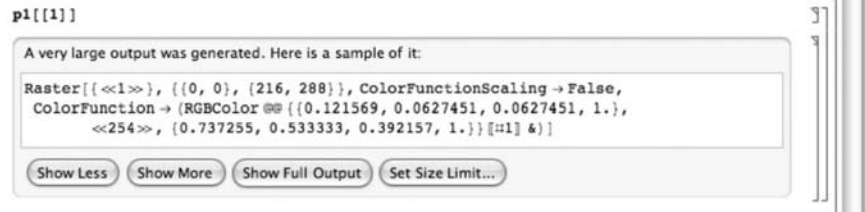
```
p1 = Import["martha01sa . gif"]
```

With `Length`, we see that `p1` has four elements. We can examine the entries with `Part`.

```
Length[p1]
```

```
4
```

With `p1[[1]]`, we select the first part of `p1`. When large output is the result of a calculation, Mathematica warns you before displaying it. (Note that `Short[p1[[1]]]` returns a similar result.)



The result helps us understand the structure of `p1` and `p1[[1]]`, which is another list. The second element of `p1` tells us the size of the image.

p1[[2]]

`ImageSize` $\rightarrow$ `{216, 288}`

The third and fourth elements tell us how the image is to be plotted—its plot range and background.

p1[[3]]

`PlotRange` $\rightarrow$ `{{0, 216}, {0, 288}}`

p1[[4]]

`Background` $\rightarrow$ `None`

The data determining the image is contained in `p1[[1,1]]`, which is a 288×216 matrix. To see so, first enter `p1[[1,1]]`



and then click on **Show More** twice.

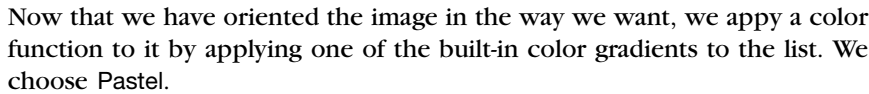
To determine the dimensions of the matrix use `Length`.

We now use `ArrayPlot` to graph `p1[[1,1]]`. `ArrayPlot` goes from up (first row) to down (last row), so our initial image (Figure 5.31(b)) is a reverse of the original.

g1a = ArrayPlot[p1[[1, 1]]]

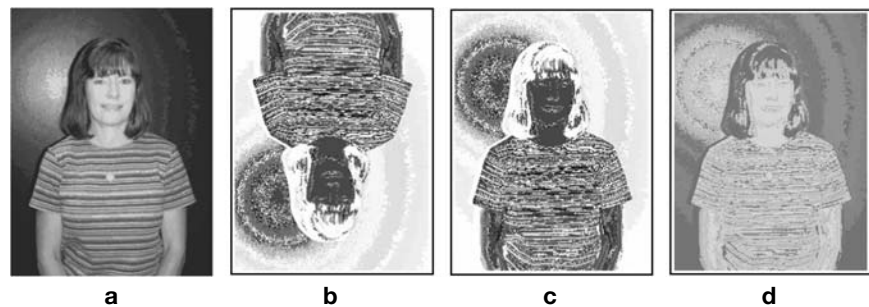
We use `Reverse` to correct the situation (Figure 5.31(c)). Generally, `Reverse[{a1, a2, ..., an}]` returns the list `{an, ..., a2, a1}`; the reverse of the original list.

g1b = ArrayPlot[Reverse[p1[[1, 1]]]]

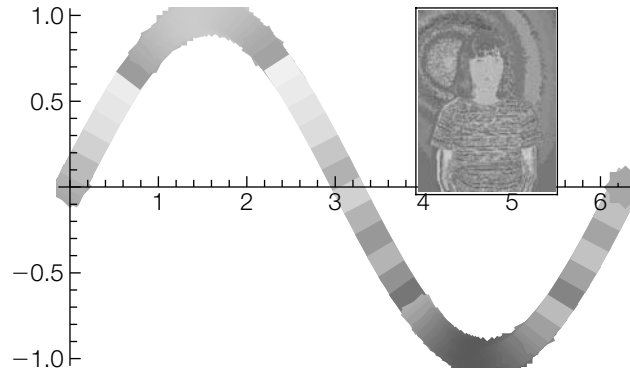


The results are shown side-by-side in Figure 5.31. Printed on a color printer, the results are amazing.

Show[GraphicsRow[{p1, g1a, g1b, g2a}]]



(a) The original. (b) Applying `ArrayPlot` to the original data points. (c) Reorienting the image. (d) Applying a color function to the data points

**FIGURE 5.32**

Use Inset to place one graphic within another

Now that we understand how to manipulate the gif image, we can be creative. In the following, the image is scaled so that the width of the image is 70 pixels (because of `ImageSize->70`). We then display the small image with another graphic. Using `Inset`, we put Martha next to a sine graph that is plotted using the same coloring gradient. See Figure 5.32.

```
g1 = ArrayPlot[p1[[1, 1]], ColorFunction -> "BrightBands",  
              ImageSize -> 70];  
p2 = Plot[Sin[x], {x, 0, 2Pi}, Epilog -> Inset[g1, {3Pi/2, 1/2}],  
          ColorFunction -> "BrightBands", PlotStyle -> Thickness[.05]]
```

An alternative way to visualize the data is to use `ListContourPlot`. To ensure that the aspect ratio of the original image is preserved, include the `AspectRatio->Automatic` option in the `ListContourPlot` command. (See Figure 5.33)

```
glp1 = ListContourPlot[p1[[1, 1]], AspectRatio -> Automatic];  
g1p2 = ListContourPlot[p1[[1, 1]], ColorFunction -> "Pastel",  
                      AspectRatio -> Automatic]  
g1p3 = ListContourPlot[p1[[1, 1]], ColorFunction -> "GrayTones",  
                      AspectRatio -> Automatic]  
Show[GraphicsRow[{glp1, g1p2, g1p3}]]
```

The underlying format of each structure (jpeg, gif, etc.) is different. With

```
p1 = Import["jim01a.jpg"];  
Length[p1]
```

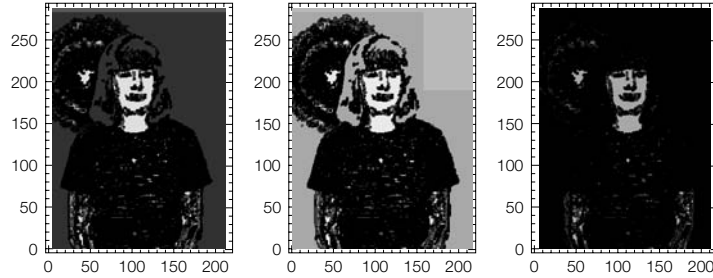
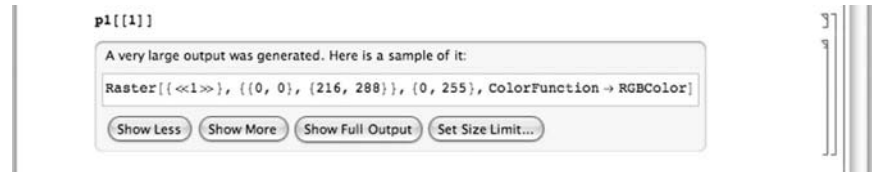


FIGURE 5.33

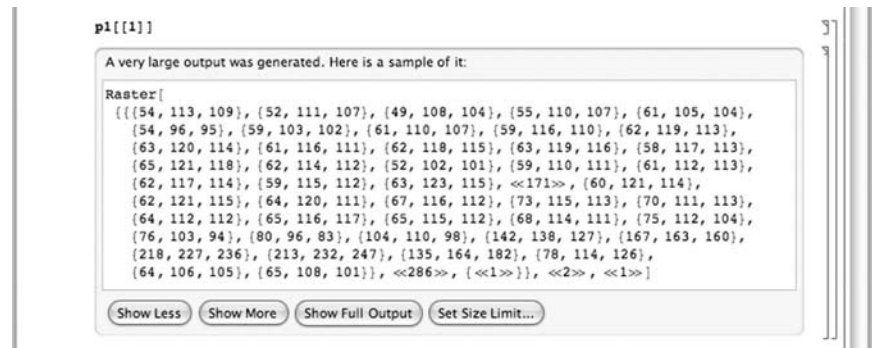
Using ListContourPlot rather than ArrayPlot

we import a jpeg of the second author of the text into Mathematica and name the result p1. Length shows us that p1 is a list with three elements. (See Figure 5.34)

The first part of p1, obtained with p1[[1]], is quite long.

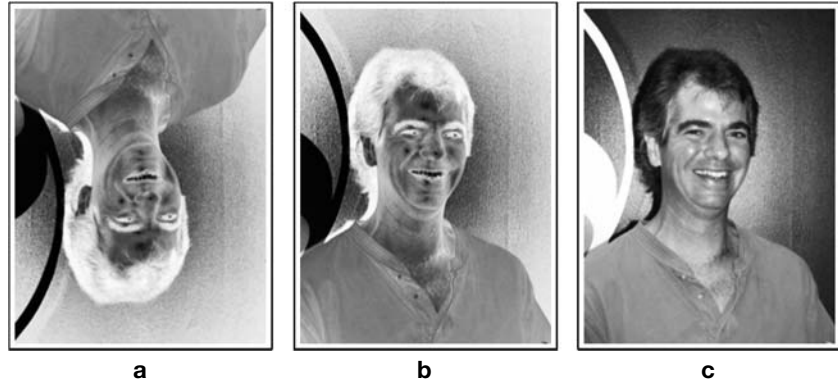


By clicking on **Show More** we see that p1[[1]] is an array of ordered triples.



The other parts of p1 are not as long. They specify the style of the image.

```
p1[[1,2]]
{{0,0},{216,288}}
p1[[1,3]]
{0,255}
p1[[1,4]]
```

**FIGURE 5.34**

(a) An upside down scary Jim. (b) Correct side up but still scary. (c) Color applied but a bit less scary

ColorFunction → RGBColor

p1[[2]]

ImageSize → {216, 288}

p1[[3]]

PlotRange → {{0, 216}, {0, 288}}

For the color jpeg, the first element of the first component is the data array that determines the image.

Short[p1[[1, 1]]]

```
{{{54, 113, 109}, {52, 111, 107}, {49, 108, 104}, {55, 110, 107},
  <<209>>, {78, 114, 126}, {64, 106, 105}, {65, 108, 101}}, <<287>>}}
```

However, as before, the image generated with ArrayPlot is upside down—and the coloring is off.

j1a = ArrayPlot[p1[[1, 1]]]

To invert the image, we use Reverse. As stated previously, Reverse[list] reverses the entries of list.

j1b = ArrayPlot[Reverse[p1[[1, 1]]]]

To correct the color, we tell Mathematica to use the RGBColor function.

j1c = ArrayPlot[Reverse[p1[[1, 1]]], ColorFunction → RGBColor]
Show[GraphicsRow[{j1a, j1b, j1c}]]

To apply your own color function, you need to manipulate the data. For this image, viewing it as a matrix, it has 288 rows and 216 columns.

```
Length[p1[[1,1]]]
288
Length[p1[[1,1,1]]]
216
```

Each entry of the matrix `p1[[1,1]]` is an ordered triple. To apply a color function to the ordered triple, we can proceed in a variety of ways. One approach is to convert the matrix to a list of ordered triples.

```
p2 = Flatten[p1[[1, 1]], 1];
Short[p2]
{{54, 113, 109}, {52, 111, 107}, {49, 108, 104}, {55, 110, 107},
<(62200)>, {63, 67, 96}, {41, 45, 74}, {40, 45, 74}, {37, 42, 71}}
```

`p2` is a list of ordered triples. Our function, *b*, adds the last two elements of each triple and divides by two. We apply *b* to `p2` with `Map` and name the result `p3`. We convert `p3` back to a 288×216 array using `Partition`. We use `ArrayPlot` to visualize the result. In this case, the gray level used to shade each cell is scaled by the corresponding entry of `p4`.

```
h[{x_, y_, z_}] = (y + z)/2;
p3 = Map[h, p2];
p4 = Partition[p3, 216];
j2a = ArrayPlot[Reverse[p4]];
```

The built-in color gradients (refer to the `ColorSchemes` palette) are functions of a single variable. Thus,

```
j2b = ArrayPlot[Reverse[p4], ColorFunction -> "SolarColors"]
```

applies the `SolarColors` function to the array (Figure 5.35(b)) whereas,

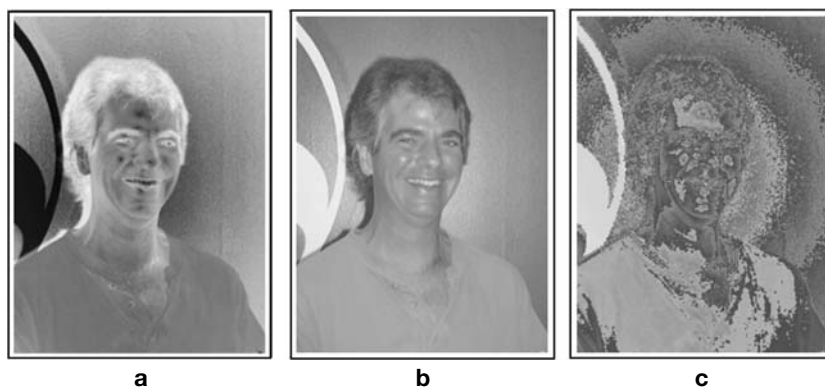
```
j2c = ArrayPlot[Reverse[p4], ColorFunction -> "DarkBands"]
```

applies `DarkBands` to the array (Figure 5.35(c)).

```
Show[GraphicsRow[{j2a, j2b, j2c}]]
```

The structure of a black-and-white jpeg differs from that of a color one. To see so, we import a *very old* picture of the second author of this text, and name the result `p1`. With `Length`, we see that `p1` has three parts

```
Length[p1]
3
```

**FIGURE 5.35**

Manipulating a color jpeg with ColorFunction

```
p1 = Import["littlejim.jpg"];
Show[p1, ImageSize -> Small]
```



`p1[[1]]` is also a list.

```
p1[[1, 1]]
```

A very large output was generated. Here is a sample of it:

```
{(244, 193, 189, 174, 182, 173, 182, 173, 171, 164, 169, 169, 162, 169,
  174, 160, 166, 168, 164, 176, 174, 171, 157, 160, 169, 156, 161,
  176, 155, 160, 165, 162, 162, <<362>>, 66, 62, 61, 62, 66, 77, 82,
  85, 100, 108, 110, 128, 139, 149, 150, 151, 153, 157, 162, 162, 160,
  137, 149, 124, 50, 88, 109, 132, 144, 155, 183, 197, 207}, <<599>>}
```

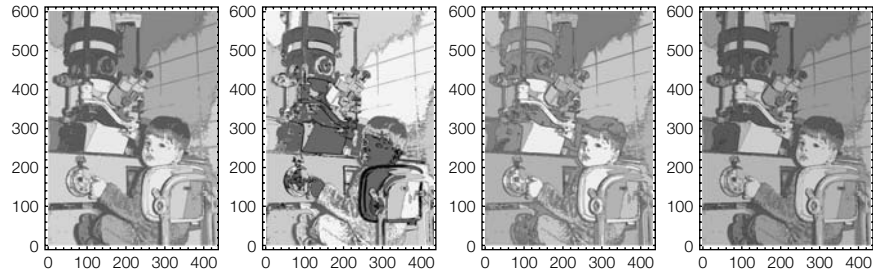
Show Less

Show More

Show Full Output

Set Size Limit...

The data defining the graphic is contained in the list `p1[[1,1]]`, which is a 600×428 array/matrix. The other parts of `p1` describe the remaining parts of the graphic.

**FIGURE 5.36**

Using `ListContourPlot` along with various options to graphically represent a matrix

```
p1[[1,1]]
p1[[2]]
ImageSize → {428, 600}
p1[[3]]
PlotRange → {{0, 428}, {0, 600}}
```

In Figure 5.36, we illustrate the use of `ListContourPlot` along with various options.

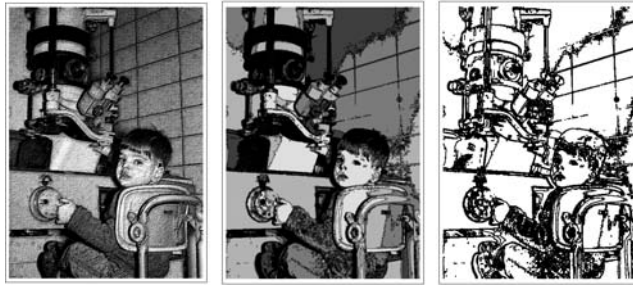
```
g1 = ListContourPlot[p1[[1, 1]], AspectRatio → Automatic]
g2 = ListContourPlot[p1[[1, 1]],
    AspectRatio → Automatic, ColorFunction → "ThermometerColors"]
g3 = ListContourPlot[p1[[1, 1]], AspectRatio → Automatic,
    ColorFunction → "DarkBands"]
g4 = ListContourPlot[p1[[1, 1]], AspectRatio → Automatic,
    ColorFunction → "SolarColors"]
Show[GraphicsGrid[{{g1, g2}, {g3, g4}}]]
```

Figure 5.37 shows variations obtained with `ReliefPlot` and `ListContourPlot`.

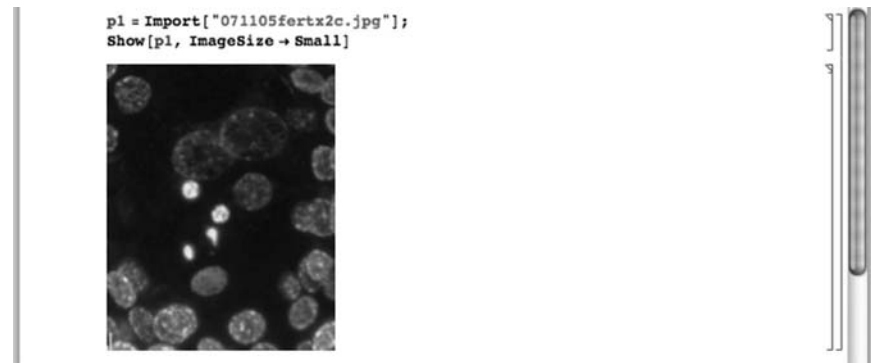
```
g2 = ReliefPlot[p1[[1, 1]], AspectRatio → Automatic,
    ColorFunction → "GrayTones", FrameTicks → None];
g3 = ListContourPlot[p1[[1, 1]], AspectRatio → Automatic,
    ColorFunction → "GrayTones", FrameTicks → None];
g4 = ListContourPlot[p1[[1, 1]], AspectRatio → Automatic,
    ContourStyle → Black, ContourShading → False,
    FrameTicks → None];
Show[GraphicsRow[{g2, g3, g4}]]
```

`ReliefPlot` can help add insight to images, especially when they have geographical or biological meaning. For example, this jpeg shows the beginning of a biological process of a cell.

```
p1 = Import["071105fertx2c.jpg"];
Show[p1, ImageSize → Small]
```

**FIGURE 5.37**

Manipulating an image with ListContourPlot and ReliefPlot

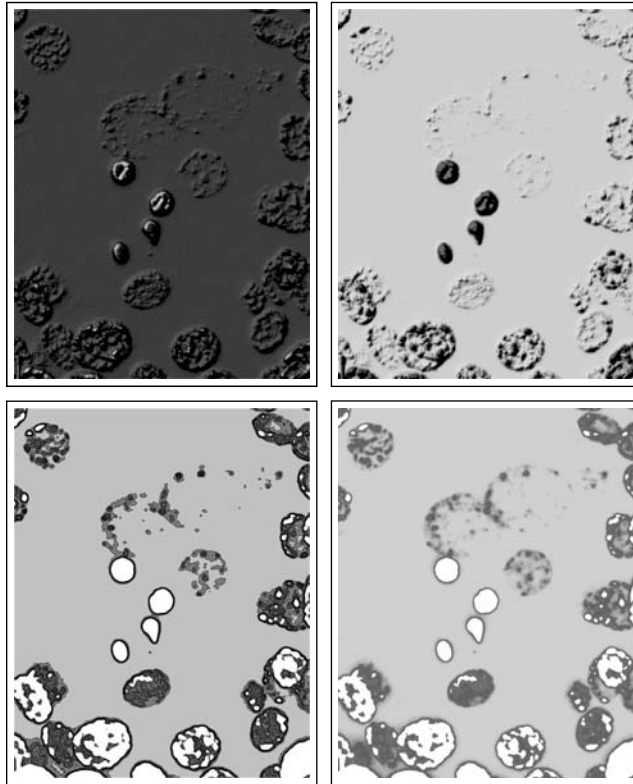


With Length, we see that $p1[[1,1]]$ is a 500×400 array.

```
Length[p1[[1,1]]]
500
Length[p1[[1,1]]]
400
```

Viewing $p1[[1,1]]$ as a 500×400 array, each entry is a 1×3 array/vector. To easily apply a function, f , that assigns a number to each ordered triple, we use Flatten to convert the nested list/array $p1[[1,1]]$ to a list of ordered triples in $p2$.

```
p2 = Flatten[p1[[1, 1]], 1];
Short[p2]
Length[p2]
{{194, 215, 158}, {189, 208, 144},
 <<199996>>, {137, 66, 46}, {139, 68, 48}}
200000
```

**FIGURE 5.38**

Using `ReliefPlot`, `ListContourPlot`, and `ListDensityPlot` along with various options to graphically represent a matrix

To apply our own color function to this data set, we convert the ordered triples to some other form. For illustrative purposes, we convert each ordered triple (x, y, z) in `p2` to the number $x + y^2$. The result is converted back to a 500×400 array, with `Partition` in `p3`.

```
f[y_]:=y[[1]] + y[[2]] ^ 2
p3 = Partition[Map[f, p2], 400];
Length[p3]
500
```

We then use `ReliefPlot`, `ListContourPlot`, and `ListDensityPlot` along with various options to graph the result in Figure 5.38.

```
g1 = ReliefPlot[p3, AspectRatio -> Automatic,
  ColorFunction -> "DarkRainbow"];
```

```

g2 = ReliefPlot[p3, AspectRatio → Automatic,
  ColorFunction → "NeonColors", Ticks → None,
  Axes → None, FrameTicks → None]
g3 = ListContourPlot[p3, AspectRatio → Automatic,
  ColorFunction → "NeonColors", Ticks → None,
  Axes → None, FrameTicks → None]
g4 = ListDensityPlot[p3, AspectRatio → Automatic,
  ColorFunction → "NeonColors", Ticks → None,
  Axes → None, FrameTicks → None]

```

5.7 EXERCISES

1. Solve $-3y - z - 3w = -1$, $-3x + 3y - 3z - 3w = -1$, $2x + 2y - z + w = 2$.
2. Find the eigenvalues and eigenvectors of each matrix. Verify that your

results are correct. (a) $B = \begin{pmatrix} 0 & 4 \\ 2 & -2 \end{pmatrix}$, (b) $A = \begin{pmatrix} 3 & 5 & -4 \\ -5 & 6 & 3 \\ -3 & 2 & -2 \end{pmatrix}$, and (c)

$$A = \begin{pmatrix} 5 & 2/3 & 1 & -4/3 & -4 & -4/3 \\ 0 & -1/6 & -2 & -1/6 & 7 & 23/6 \\ -1/2 & -1/4 & 5/2 & -1/4 & 1 & 3/4 \\ 4 & 1/2 & 0 & 1/2 & -3 & -1/2 \\ 0 & -1 & 0 & 0 & 4 & 1 \\ -1 & 1/6 & -1 & 1/6 & 2 & 19/6 \end{pmatrix} \quad \text{Comment: In some cases,}$$

numerical results (use N) may be more meaningful than the exact ones.

3. For each of the following matrices, find the eigenvalues, eigenvectors, characteristic polynomial, and minimal polynomial:

$$(a) A = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \quad (b) B = \begin{pmatrix} -3 & -2 & -2 & -4 \\ 2 & 1 & 2 & 4 \\ 2 & 2 & 1 & 4 \\ -1 & -1 & -1 & -3 \end{pmatrix},$$

$$(c) C = \begin{pmatrix} -3 & -1 & 5 & -5 \\ -6 & -9 & 10 & -15 \\ 6 & 8 & -11 & 15 \\ 8 & 10 & -14 & 19 \end{pmatrix}, \quad \text{and (d) } D = \begin{pmatrix} -3 & 0 & 1 & -1 \\ -1 & -1 & 0 & -1 \\ -3 & 1 & 1 & -1 \\ 0 & 1 & 0 & -1 \end{pmatrix}.$$

4. Let $J_n(\lambda)$ denote the $n \times n$ matrix with λ 's down the diagonal, 0's below, and 1's to the right of each λ (for example, $J_3(\lambda) =$

$$\begin{pmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{pmatrix}), \quad \text{and let } J = \begin{pmatrix} J_n(\lambda_1) & 0 \\ 0 & J_m(\lambda_2) \end{pmatrix} \text{ denote the } (n+m) \times (n+m)$$

matrix with “blocks” $\mathbf{J}_n$ and $\mathbf{J}_m$ and 0's elsewhere (for example,

$$\begin{pmatrix} \mathbf{J}_2(\lambda_1) & 0 \\ 0 & \mathbf{J}_3(\lambda_2) \end{pmatrix} = \begin{pmatrix} \lambda_1 & 1 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2 & 1 & 0 \\ 0 & 0 & 0 & \lambda_2 & 1 \\ 0 & 0 & 0 & 0 & \lambda_2 \end{pmatrix}).$$
 Find the eigenvalues, eigen-

vectors, characteristic polynomial, and minimal polynomial of $\mathbf{J}$. Illustrate your results with examples.

5. Find the unit normals to $w = \cos(4x^2 + 9y^2)$. Illustrate the result graphically.
6. Evaluate $\oint_C (e^{\sqrt{y}} + x) dx + (2y + \cos x) dy$, where C is the boundary of the region between $y = x^2$ and $x = y^2$.
7. Find the outward flux of the vector field

$$\mathbf{F}(x, y, z) = (xy + x^2yz) \mathbf{i} + (yz + xy^2z) \mathbf{j} + (xz + xyz^2) \mathbf{k}$$

through the surface of the cube cut from the first octant by the planes $x = 2$, $y = 2$, and $z = 2$.

8. Verify Stokes' theorem for the vector field

$$\mathbf{F}(x, y, z) = (y^2 - z) \mathbf{i} + (x + z^2) \mathbf{j} + (x^2 - y) \mathbf{k}$$

and S the portion of the paraboloid $z = f(x, y) = 4 - (x^2 + y^2)$, $z \geq 0$.

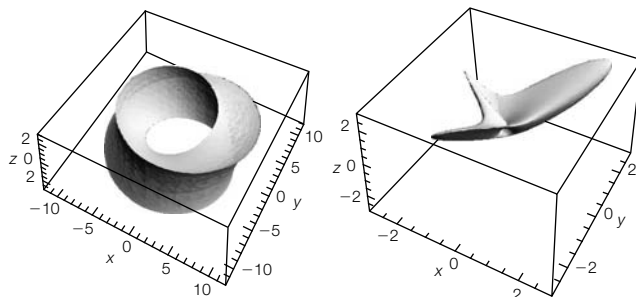
9. The **Fibonacci numbers** satisfy the recurrence relation $F_n = F_{n-2} + F_{n-1}$, where $F_1 = F_2 = 1$. Provide a convincing argument that

$$\begin{vmatrix} F_{n+1} & F_{n+2} & \cdots & F_{n+k} \\ F_{n+k+1} & F_{n+k+2} & \cdots & F_{n+2k} \\ \vdots & \vdots & \vdots & \vdots \\ F_{n+k(k-1)+1} & F_{n+k(k-1)+2} & \cdots & F_{n+k^2} \end{vmatrix} = 0.$$

Suggestion: Use Fibonacci.

10. The **Boy surface** has parametrization

$$\begin{aligned} x(s, t) &= \frac{\sqrt{2} \cos^2 t \cos 2s + \cos s \sin 2t}{2 - \sqrt{2} \sin 3s \sin 2t} \\ y(s, t) &= \frac{\sqrt{2} \cos^2 t \cos 2s + \cos s \sin 2t}{2 - \sqrt{2} \sin 3s \sin 2t} \\ z(s, t) &= \frac{3 \cos^2 t}{2 - \sqrt{2} \sin 3s \sin 2t} \end{aligned}$$

**FIGURE 5.39**

On the left, umbilic torus; on the right Boy surface

and the **umbilic torus** has parametrization

$$x(s, t) = (7 + \cos(s/3 - 2t) + 2 \cos(s/3 + t)) \sin s$$

$$y(s, t) = (7 + \cos(s/3 - 2t) + 2 \cos(s/3 + t)) \cos s$$

$$z(s, t) = \sin(s/3 - 2t) + 2 \sin(s/3 + t).$$

See Figure 5.39. Determine if either of these surfaces is orientable.

11. Using 0's for dots and 1's for dashes and omitting spaces and punctuation, the following phrases are translated to Morse code as follows: "S.O.S" becomes 0, 0, 0, 1, 1, 1, 0, 0, 0; "Save our souls" becomes 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0; "Mathematica is terrific" becomes 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0; "Can I borrow the car" becomes 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0; "Are aliens on earth" becomes 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0; and "Work harder" becomes 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0. Represent this array graphically using (at least) three different methods. *Challenge:* Write a function `symboltomorse` that converts strings of letters to Morse code. Translate your favorite five quotes into Morse code and represent the result graphically.
12. A given curvature function determines a plane curve: The curve C parametrized by arc length with curvature $\kappa(s)$ has parametrization $\mathbf{r}(s) = \langle x(s), y(s) \rangle$, where

$$\begin{cases} dx/ds = \cos \theta \\ dy/ds = \sin \theta \\ d\theta/ds = \kappa \\ x(a) = c, y(a) = d, \theta(0) = \theta_0 \end{cases} \quad (5.27)$$

You can often use `NDSolve` to solve system (5.27). Plot the curve C for which $\kappa(s) = e^{-s} + e^s$ for $-5 \leq s \leq 5$ if $x(0) = y(0) = \theta(0) = 0$.
Hint:

```
In[4]:= Clear[curvek, κ];
curvek[k_, ss_: {s, -15, 15}, opts___] :=
Module[{numsol},
  numsol = NDSolve[{x'[s] == Cos[θ[s]], y'[s] == Sin[θ[s]], θ'[s] == k,
    x[0] == 0, y[0] == 0, θ[0] == 0}, {x[s], y[s], θ[s]}, ss];
  ParametricPlot[Evaluate[{x[s], y[s]} /. numsol], ss, opts,
    AspectRatio -> Automatic]
]
```

Repeat the exercise given the following curvature functions (all for $-40 \leq s \leq 40$): $\kappa(s) = s + \sin s$, $\kappa(s) = sJ_1(s)$, $\kappa(s) = sJ_2(s)$, $\kappa(s) = s \sin \sin(s)$, $\kappa(s) = s \sin^2 \sin(s)$, and $\kappa(s) = |s \sin \sin(s)|$. See Figure 5.40.

13. Consider placing a tube around a curve (refer to `tubeplot`) but letting the radius change:

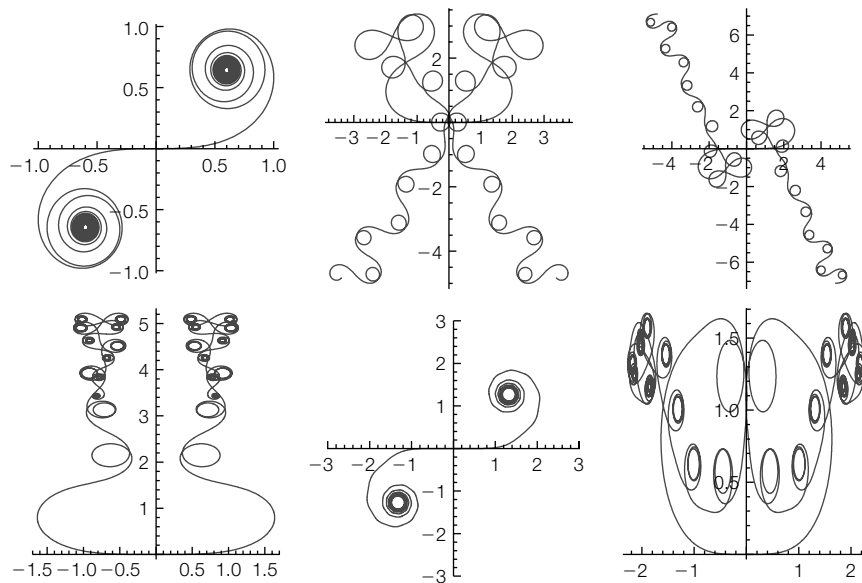


FIGURE 5.40

You can generate stunning curves by specifying a curvature function

seashell[γ][r][t, θ] := $\gamma[t] +$
 $r t (\text{Cos}[\theta] \text{normal}[\gamma][t] + \text{Sin}[\theta] \text{binormal}[\gamma][t])$

- (a) Use **seashell** with $\gamma(t) = \langle t, t \cos 2t, t \sin 2t \rangle$ to create Figure 5.41.
 (b) Illustrate the curve with its Frenet frame field.

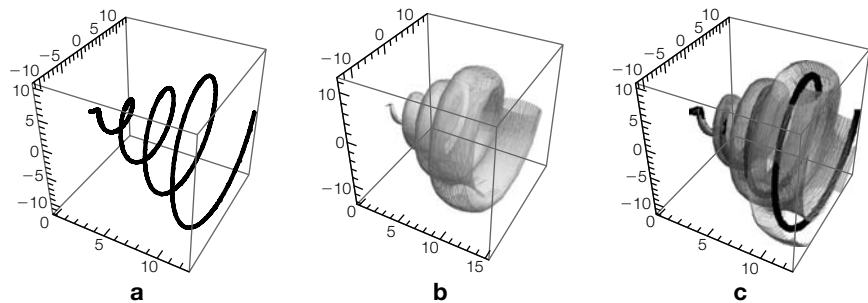


FIGURE 5.41

(a) A winding curve. (b) A seashell. (c) A winding curve in a seashell

CHAPTER Applications Related to Ordinary and Partial Differential Equations

6

For more detailed discussions regarding Mathematica and differential equations, see references such as Abell and Braselton's *Differential Equations with Mathematica* [1].

Chapter 6 discusses Mathematica's differential equations commands. The examples used to illustrate the various commands are similar to examples routinely done in a one- or two-semester differential equations course.

6.1 FIRST-ORDER DIFFERENTIAL EQUATIONS

6.1.1 Separable Equations

Because they are solved by integrating, separable differential equations are usually the first introduced in the introductory differential equations course.

Definition 1 (Separable Differential Equation). *A differential equation of the form*

$$f(y) dy = g(t) dt \quad (6.1)$$

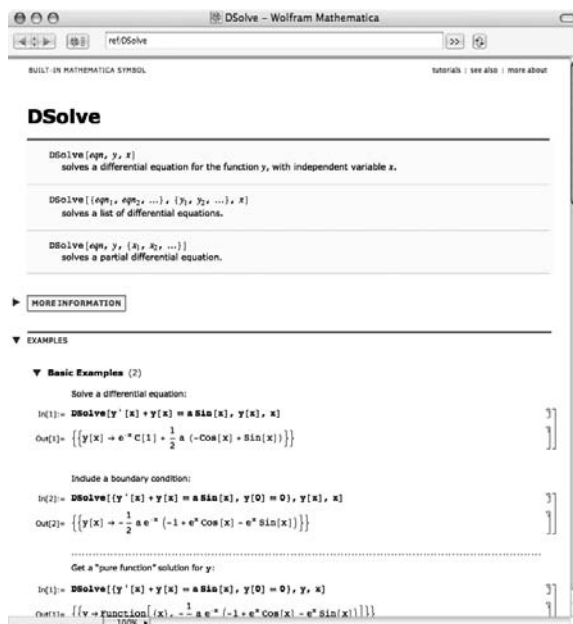
is called a first-order separable differential equation.

We solve separable differential equations by integrating.

Remark 6.1 The command

`DSolve[y'[t]==f[t, y[t]], y[t], t]`

attempts to solve $y' = dy/dt = f(t, y)$ for y .



Example 6.1.1 Solve each of the following equations: (a) $y' - y^2 \sin t = 0$; (b) $y' = \alpha y \left(1 - \frac{1}{K}y\right)$, $K, \alpha > 0$ constant.

Solution (a) The equation is separable so we separate and then integrate:

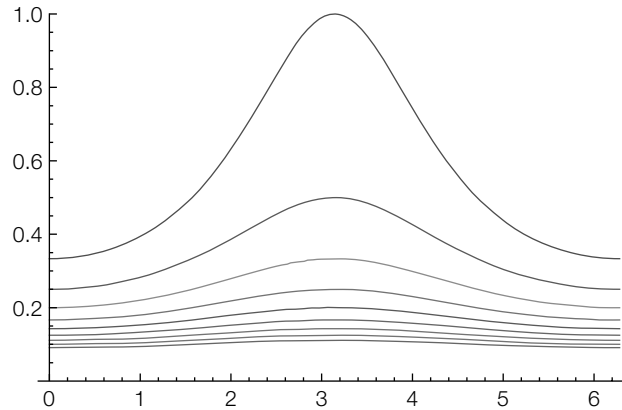
$$\begin{aligned}\frac{1}{y^2} dy &= \sin t \, dt \\ \int \frac{1}{y^2} dy &= \int \sin t \, dt \\ -\frac{1}{y} &= -\cos t + C \\ y &= \frac{1}{\cos t + C}.\end{aligned}$$

We check our result with DSolve.

$$\begin{aligned}\text{sola} &= \text{DSolve}[y'[t] - y[t]^2 \sin[t] == 0, y[t], t] \\ &= \left\{ \left\{ y[t] \rightarrow \frac{1}{-C[1] + \cos[t]} \right\} \right\}\end{aligned}$$

Observe that the result is given as a list. The formula for the solution is the second part of the first part of the first part of **sola**.

$$\begin{aligned}\text{sola}[[1, 1, 2]] \\ &= \frac{1}{-C[1] + \cos[t]}\end{aligned}$$

**FIGURE 6.1**

Several solutions of $y' - y^2 \sin t = 0$

We then graph the solution for various values of C with `Plot` in Figure 6.1.

`toplota = Table[sola[[1, 1, 2]]/.C[1] → -i, {i, 2, 10}]`

`{` $\left\{ \frac{1}{2 + \cos[t]}, \frac{1}{3 + \cos[t]}, \frac{1}{4 + \cos[t]}, \frac{1}{5 + \cos[t]}, \frac{1}{6 + \cos[t]}, \frac{1}{7 + \cos[t]}, \frac{1}{8 + \cos[t]}, \frac{1}{9 + \cos[t]}, \frac{1}{10 + \cos[t]} \right\}$ **`}`**

**`Plot[Tooltip[toplota], {t, 0, 2Pi}, PlotRange → {0, 1},
AxesOrigin → {0, 0}]`**

expression /. x->y
replaces all
occurrences of x in
expression by y .
`Table[a[k], {k, n, m}]`
generates the list a_n ,
 $a_{n+1}, \dots, a_{m-1}, a_m$.

To graph the list of
functions `{list}` for
 $a \leq x \leq b$, enter
`Plot[list, {x, a, b}]`.

(b) After separating variables, we use partial fractions to integrate:

$$y' = \alpha y \left(1 - \frac{1}{K} y \right)$$

$$\frac{1}{\alpha y \left(1 - \frac{1}{K} y \right)} dy = dt$$

$$\frac{1}{\alpha} \left(\frac{1}{y} + \frac{1}{K - y} \right) = dt$$

$$\frac{1}{\alpha} (\ln |y| - \ln |K - y|) = C_1 + t$$

$$\frac{y}{K - y} = C e^{\alpha t}$$

$$y = \frac{C K e^{\alpha t}}{C e^{\alpha t} - 1}.$$

We check the calculations with Mathematica. First, we use `Apart` to find the partial fraction decomposition of $\frac{1}{\alpha y \left(1 - \frac{1}{K} y \right)}$.

`s1 = Apart[1/(α y (1 - 1/k y)), y]`

$$\frac{1}{y\alpha} - \frac{1}{(-k+y)\alpha}$$

Then, we use `Integrate` to check the integration.

```
s2 = Integrate[s1, y]
k ( Log[y] / (k α) - Log[-k + y] / (k α) )
```

Last, we use `Solve` to solve $\frac{1}{\alpha} (\ln |y| - \ln |K - y|) = ct$ for y .

```
Solve[s2==c + t, y]
{ { y -> e^{c α + t α k} / (-1 + e^{c α + t α k}) } }
```

We can use `DSolve` to find a general solution of the equation

```
solb = DSolve[y'[t]==α y[t](1 - 1/ky[t]), y[t], t]
{ { y[t] -> e^{t α + k C[1] k} / (-1 + e^{t α + k C[1]}) } }
```

as well as find the solution that satisfies the initial condition $y(0) = y_0$, although Mathematica generates several error messages because inverse functions are being used so the resulting solution set may not be complete.

```
solc = DSolve[{y'[t]==α y[t](1 - 1/ky[t]), y[0]==y0}, y[t], t]
```

`Solve::ifun` : Inverse functions are being used by `Solve`, so some solutions may not be found; use `Reduce` for complete solution information.}}

$$\left\{ \left\{ y[t] \rightarrow \frac{e^{t\alpha} k y_0}{k - y_0 + e^{t\alpha} y_0} \right\} \right\}$$

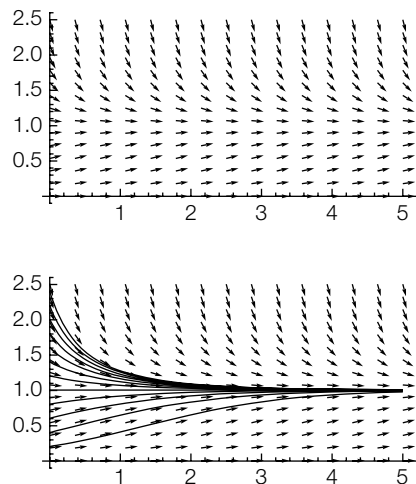
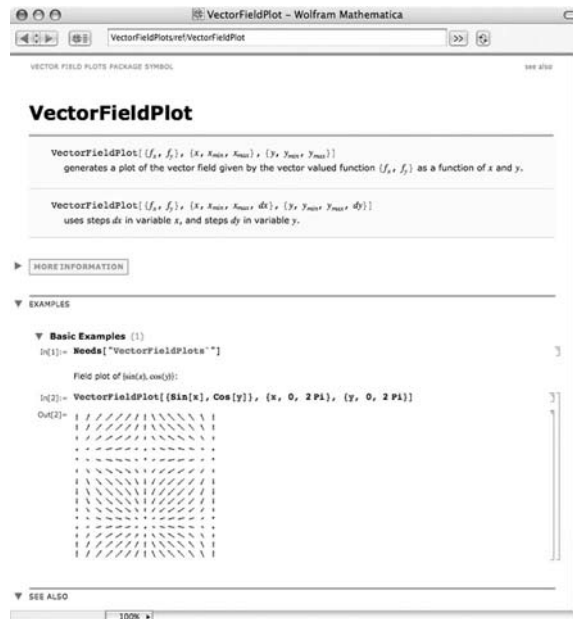
The equation $y' = \alpha y (1 - \frac{1}{K} y)$ is called the **logistic equation** (or **Verhulst equation**) and is used to model the size of a population that is not allowed to grow in an unbounded manner. Assuming that $y(0) > 0$, then all solutions of the equation have the property that $\lim_{t \rightarrow \infty} y(t) = K$.

To see this, we set $\alpha = K = 1$ and use `VectorFieldPlot`, which is contained in the **VectorFieldPlots** package, to graph the direction field associated with the equation in Figure 6.2.

```
Needs["VectorFieldPlots"]
pvf1 = Show[VectorFieldPlot[{1, y(1 - y)}, {t, 0, 5}, {y, 0, 5/2},
  ScaleFunction -> (1 &)], Axes -> Automatic, AxesOrigin -> {0, 0}];
```

The property is more easily seen when we graph various solutions along with the direction field as done next in Figure 6.2.

```
toplot = Table[solc[[1, 1, 2]]/.{α -> 1, k -> 1, y0 -> i/5}, {i, 1, 12}];
sols = Plot[toplot, {t, 0, 5}, PlotStyle -> GrayLevel[0], PlotRange -> All];
pvf2 = Show[pvf1, sols];
Show[GraphicsColumn[{pvf1, pvf2}]]
```

**FIGURE 6.2**

(Top) A typical direction field for the logistic equation. (Bottom) A typical direction field for the logistic equation along with several solutions

When Mathematica encounters inverse functions, it might choose the incorrect *branch* to form a continuous solution to an initial-value problem.

Example 6.1.2 Solve $dy/dt = \sin t \cos y$, $y(1) = 3$.

Solution When we use DSolve to solve the equation and the initial-value problem, Mathematica warns us that inverse functions are being used.

```
sol = DSolve[y' [t] == Sin[t] Cos[y[t]], y[t], t]
Solve::ifun: Inverse functions are being used by Solve, so
some solutions may not be found; use Reduce for complete solution information. >>
{{y[t] -> 2 ArcTan[Tanh[1/4 (C[1] - 2 Cos[t])]]}}
```

```
sol1 = DSolve[{y' [t] == Sin[t] Cos[y[t]], y[1] == 3}, y[t], t]
Solve::ifun: Inverse functions are being used by Solve, so
some solutions may not be found; use Reduce for complete solution information. >>
Solve::ifun: Inverse functions are being used by Solve, so
some solutions may not be found; use Reduce for complete solution information. >>
{{y[t] -> 2 ArcTan[Tanh[1/4 (2 (2 ArcTan[Tan[3/2]] + Cos[1]) - 2 Cos[t])]]}}
```

From the direction field, we see that the solution satisfying $y(1) = 3$ is continuous for (at least) $0 \leq t \leq 4\pi$. However, the explicit solution returned by DSolve is not the solution that is continuous on $[0, 4\pi]$. See Figure 6.3(a).

Needs["VectorFieldPlots"]

```
pvf1 = Show[VectorFieldPlot[{1, Sin[t]Cos[y]}, {t, 0, 4Pi}, {y, -2Pi, 2Pi},
ScaleFunction -> {1&}, PlotPoints -> 25], Axes -> Automatic,
AxesOrigin -> {0, 0}];
```

```
psol1 = Plot[y[t]/.sol1, {t, 0, 4Pi},
PlotStyle -> {{GrayLevel[.5], Thickness[.01]}}];
discont = Show[pvf1, psol1]
```

To see the continuous solution, we use NDSolve to generate a numerical solution to the initial value problem. If possible,

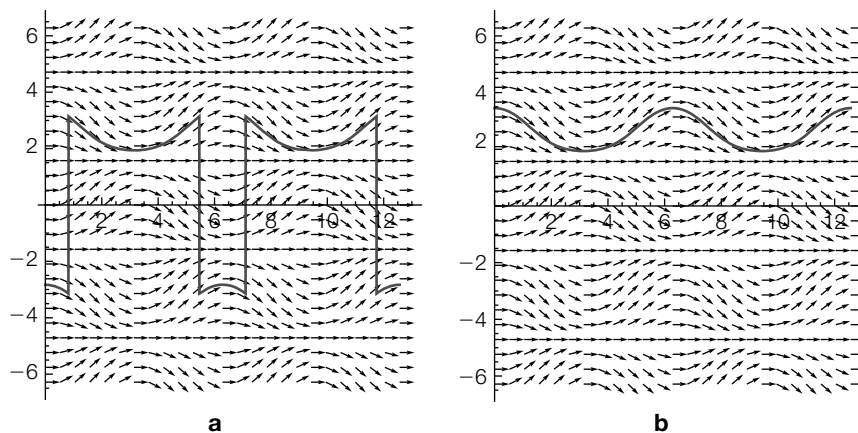
NDSolve is discussed in more detail later in the chapter.

NDSolve[{y'[t]==f[t,y[t]],y[t0]=y0},y[t],{t,a,b}] attempts to numerically solve $y' = f(t, y)$, $y(t_0) = y_0$ for $a \leq t \leq b$.

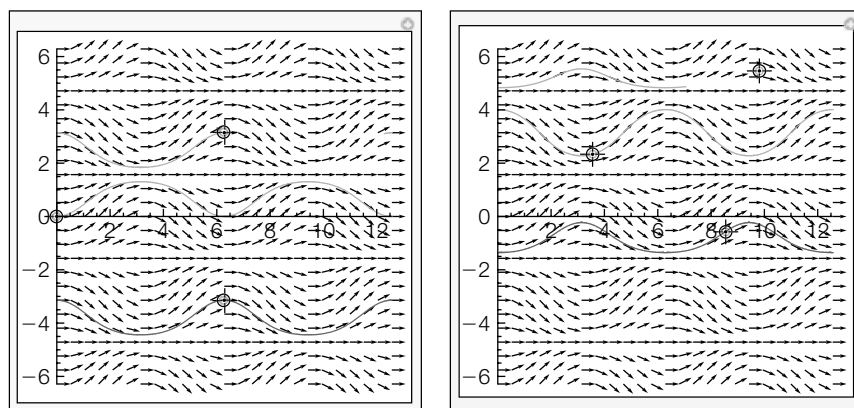
```
sol2 = NDSolve[{y'[t]==Sin[t]Cos[y[t]], y[1]==3}, y[t], {t, 0, 4Pi}]
{{y[t] -> InterpolatingFunction[{{0., 12.5664}}, <>][t]}}
```

In Figure 6.3(b), we see that the result returned by NDSolve is continuous on $[0, 4\pi]$.

```
psol2 = Plot[y[t]/.sol2, {t, 0, 4Pi},
PlotStyle -> {{GrayLevel[.5], Thickness[.01]}}];
cont = Show[pvf1, psol2]
```

**FIGURE 6.3**

(a) The solution returned by `DSolve` is discontinuous on $[0, 4\pi]$. (b) We use `NDSolve` to find the continuous solution of the initial-value problem

**FIGURE 6.4**

Visualizing how changing t_0 and y_0 affects the solution that satisfies $y(t_0) = y_0$

With `Manipulate`, you can see how varying the initial conditions affects the solution. See Figure 6.4. When you drag the locator points, the solution changes accordingly.

```
Manipulate[
sol1 = NDSolve[{y'[t] == Sin[t] Cos[y[t]], y[pt[[1, 1]]] == pt[[1, 2]]},
          y[t], {t, 0, 4Pi}];
psol1 = Plot[y[t]/.sol1, {t, 0, 4Pi},
          PlotStyle -> {{GrayLevel[.7], Thickness[.01]}}];
```

```

sol2 = NDSolve[{y'[t]==Sin[t]Cos[y[t]], y[pt][[2, 1]]==pt[[2, 2]],
  y[t], {t, 0, 4Pi}];
psol2 = Plot[y[t]/.sol2, {t, 0, 4Pi},
  PlotStyle -> {{GrayLevel[.7], Dashing[{0.02}], Thickness[.01]}}];
sol3 = NDSolve[{y'[t]==Sin[t]Cos[y[t]], y[pt][[3, 1]]==pt[[3, 2]], y[t],
  {t, 0, 4Pi}];
psol3 = Plot[y[t]/.sol3, {t, 0, 4Pi},
  PlotStyle -> {{GrayLevel[.4], Thickness[.01]}}];
initialpt = Graphics[Point[{pt}], PlotRange -> {{0, 4Pi}, {-2Pi, 2Pi}}];
Show[pvf1, psol1, psol2, psol3, initialpt, Axes -> Automatic,
  PlotRange -> {{0, 4Pi}, {-2Pi, 2Pi}}, AspectRatio -> Automatic],
{{pt, {{0, 0}, {2Pi, Pi}, {2Pi, -Pi}}}, Locator}
]

```

6.1.2 Linear Equations

Definition 2 (First-Order Linear Equation). A differential equation of the form

$$a_1(t) \frac{dy}{dt} + a_0(t)y = f(t), \quad (6.2)$$

where $a_1(t)$ is not identically the zero function, is a first-order **linear differential equation**.

Assuming that $a_1(t)$ is not identically the zero function, dividing equation (6.2) by $a_1(t)$ gives us the **standard form** of the first-order linear equation:

$$\frac{dy}{dt} + p(t)y = q(t). \quad (6.3)$$

If $q(t)$ is identically the zero function, we say that the equation is **homogeneous**. The **corresponding homogeneous equation** of equation (6.3) is

$$\frac{dy}{dt} + p(t)y = 0. \quad (6.4)$$

Observe that equation (6.4) is separable:

$$\begin{aligned}
 \frac{dy}{dt} + p(t)y &= 0 \\
 \frac{1}{y} dy &= -p(t) dt \\
 \ln |y| &= -\int p(t) dt + C \\
 y &= Ce^{-\int p(t) dt}.
 \end{aligned}$$

A particular solution is a specific solution to the equation that does not contain any arbitrary constants.

Notice that any constant multiple of a solution to a linear homogeneous equation is also a solution. Now suppose that y is any solution of equation (6.3) and y_p is a particular solution of equation (6.3). Then,

$$\begin{aligned}(y - y_p)' + p(t)(y - y_p) &= y' + p(t)y - (y_p' + p(t)y_p) \\ &= q(t) - q(t) = 0.\end{aligned}$$

Thus, $y - y_p$ is a solution to the corresponding homogeneous equation of equation (6.3). Hence,

$$\begin{aligned}y - y_p &= Ce^{-\int p(t) dt} \\ y &= Ce^{-\int p(t) dt} + y_p \\ y &= y_b + y_p,\end{aligned}$$

where $y_b = Ce^{-\int p(t) dt}$. That is, a general solution of equation (6.3) is $y = y_b + y_p$, where y_p is a particular solution to the nonhomogeneous equation and y_b is a general solution to the corresponding homogeneous equation. Thus, to solve equation (6.3), we need to first find a general solution to the corresponding homogeneous equation, y_b , which we can accomplish through separation of variables, and then find a particular solution, y_p , to the nonhomogeneous equation.

If y_b is a solution to the corresponding homogeneous equation of equation (6.3), then for any constant C , Cy_b is also a solution to the corresponding homogeneous equation. Therefore, it is impossible to find a particular solution to equation (6.3) of this form. Instead, we search for a particular solution of the form $y_p = u(t)y_b$, where $u(t)$ is *not* a constant function. Assuming that a particular solution, y_p , to equation (6.3) has the form $y_p = u(t)y_b$, differentiating gives us $y_p' = u'y_b + uy_b'$ and substituting into equation (6.3) results in

$$y_p' + p(t)y_p = u'y_b + uy_b' + p(t)uy_b = q(t).$$

Because $uy_b' + p(t)uy_b = u[y_b' + p(t)y_b] = u \cdot 0 = 0$, we obtain

$$\begin{aligned}u'y_b &= q(t) \\ u' &= \frac{1}{y_b}q(t) \\ u' &= e^{\int p(t) dt}q(t) \\ u &= \int e^{\int p(t) dt}q(t) dt\end{aligned}$$

so

$$y_p = u(t)y_b = Ce^{-\int p(t) dt} \int e^{\int p(t) dt} q(t) dt.$$

y_b is a solution to the corresponding homogeneous equation, so $y_b' + p(t)y_b = 0$.

Because we can include an arbitrary constant of integration when evaluating $\int e^{\int p(t) dt} q(t) dt$, it follows that we can write a general solution of equation (6.3) as

$$y = e^{-\int p(t) dt} \int e^{\int p(t) dt} q(t) dt. \quad (6.5)$$

Alternatively, multiplying equation (6.3) by the **integrating factor** $\mu(t) = e^{\int p(t) dt}$ gives us the same result:

$$\begin{aligned} e^{\int p(t) dt} \frac{dy}{dt} + p(t) e^{\int p(t) dt} y &= q(t) e^{\int p(t) dt} \\ \frac{d}{dt} \left(e^{\int p(t) dt} y \right) &= q(t) e^{\int p(t) dt} \\ e^{\int p(t) dt} y &= \int q(t) e^{\int p(t) dt} dt \\ y &= e^{-\int p(t) dt} \int q(t) e^{\int p(t) dt} dt. \end{aligned}$$

Thus, first-order linear equations can always be solved, although the resulting integrals may be difficult or impossible to evaluate exactly.

Mathematica is able to solve the general form of the first-order equation, the initial-value problem $y' + p(t)y = q(t)$, $y(0) = y_0$,

$$\begin{aligned} &\text{DSolve}[y'[t] + p[t] y[t] == q[t], y[t], t] \\ &\left\{ \left\{ y[t] \rightarrow e^{\int_0^t p[K[1]] dK[1]} C[1] + \right. \right. \\ &\quad \left. \left. e^{\int_0^t p[K[1]] dK[1]} \int_0^t e^{-\int_0^K p[K[2]] dK[2]} q[K[2]] dK[2] \right\} \right\} \end{aligned}$$

$$\begin{aligned} &\text{DSolve}[\{y'[t] + p[t] y[t] == q[t], y[0] == y0\}, y[t], t] \\ &\left\{ \left\{ y[t] \rightarrow -e^{-\int_0^t p[K[1]] dK[1]} \int_0^t e^{\int_0^K p[K[2]] dK[2]} \right. \right. \\ &\quad \left. \left. (-y0 + e^{\int_0^0 p[K[1]] dK[1]} \int_0^0 e^{-\int_0^K p[K[2]] dK[2]} q[K[2]] dK[2] - \right. \right. \\ &\quad \left. \left. e^{\int_0^0 p[K[1]] dK[1]} \int_0^t e^{-\int_0^K p[K[2]] dK[2]} q[K[2]] dK[2] \right\} \right\} \end{aligned}$$

as well as the corresponding homogeneous equation,

$$\begin{aligned} &\text{DSolve}[y'[t] + p[t] y[t] == 0, y[t], t] \\ &\left\{ \left\{ y[t] \rightarrow e^{\int_0^t p[K[1]] dK[1]} C[1] \right\} \right\} \end{aligned}$$

$$\begin{aligned} &\text{DSolve}[\{y'[t] + p[t] y[t] == 0, y[0] == y0\}, y[t], t] \\ &\left\{ \left\{ y[t] \rightarrow e^{-\int_0^t p[K[1]] dK[1]} \int_0^0 e^{\int_0^K p[K[2]] dK[2]} y0 \right\} \right\} \end{aligned}$$

although the results contain unevaluated integrals.

Example 6.1.3 (Exponential Growth). Let $y = y(t)$ denote the size of a population at time t . If y grows at a rate proportional to the amount present, y satisfies

$$\frac{dy}{dt} = \alpha y, \quad (6.6)$$

where α is the **growth constant**. If $y(0) = y_0$, using equation (6.5) results in $y = y_0 e^{\alpha t}$. We use DSolve to confirm this result.

```
DSolve[{y'[t]==α y[t], y[0]==y0}, y[t], t]
{{y[t] -> e^{tα} y0}}
```

Example 6.1.4 Solve each of the following equations: (a) $dy/dt = k(y - y_s)$, $y(0) = y_0$, k and y_s constant; (b) $y' - 2ty = t$ (c) $ty' - y = 4t \cos 4t - \sin 4t$.

Solution

$dy/dt = k(y - y_s)$
models *Newtons law of cooling*: The rate at which the temperature, $y(t)$, changes in a heating/cooling body is proportional to the difference between the temperature of the body and the constant temperature, y_s , of the surroundings.

This will turn out to be a lucky guess. If there is not a solution of this form, we would not find one of this form.

(a) By hand, we rewrite the equation and obtain $y' - ky = -ky_s$. A general solution of the corresponding homogeneous equation $y' - ky = 0$ is $y_h = e^{kt}$. Because k and $-ky_s$ are constants, we suppose that a particular solution of the nonhomogeneous equation, y_p , has the form $y_p = A$, where A is a constant.

Assuming that $y_p = A$, we have $y'_p = 0$, and substitution into the nonhomogeneous equation gives us

$$y'_p - ky_p = -KA = -ky_s \quad \text{so} \quad A = y_s.$$

Thus, a general solution is $y = y_h + y_p = Ce^{kt} + y_s$. Applying the initial condition $y(0) = y_0$ results in $y = y_s + (y_0 - y_s)e^{kt}$.

We obtain the same result with DSolve. We graph the solution satisfying $y(0) = 75$ assuming that $k = -1/2$ and $y_s = 300$ in Figure 6.5. Notice that $y(t) \rightarrow y_s$ as $t \rightarrow \infty$.

```
sola = DSolve[{y'[t]==k(y[t]-ys), y[0]==y0}, y[t], t]
```

```
{{y[t] -> e^{kt} y0 + ys - e^{kt} ys}}
```

```
Plot[y[t]/.sola/.{k -> -1/2, ys -> 300, y0 -> 75}, {t, 0, 10}]
```

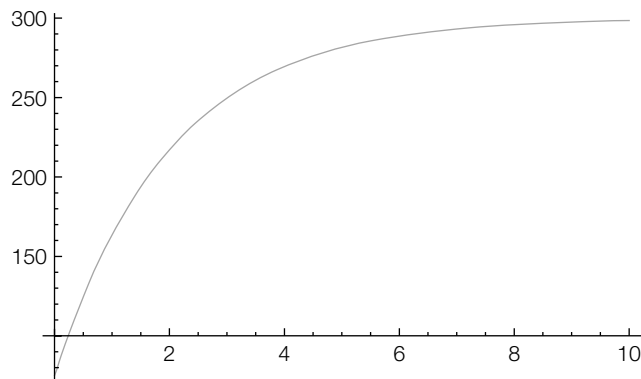
(b) The equation is in standard form and we identify $p(t) = -2t$. Then, the integrating factor is $\mu(t) = e^{\int p(t) dt} = e^{-t^2}$. Multiplying the equation by the integrating factor, $\mu(t)$, results in

$$e^{-t^2}(y' - 2ty) = te^{-t^2} \quad \text{or} \quad \frac{d}{dt}(ye^{-t^2}) = te^{-t^2}.$$

Integrating gives us $ye^{-t^2} = -\frac{1}{2}e^{-t^2} + C$ or $y = -\frac{1}{2} + Ce^{t^2}$. We confirm the result with DSolve.

```
DSolve[y'[t] - 2ty[t]==t, y[t], t]
```

```
{{y[t] -> -1/2 + e^{t^2} C[1]}}
```

**FIGURE 6.5**

The temperature of the body approaches the temperature of its surroundings

(c) In standard form, the equation is $y' - y/t = (4t \cos 4t - \sin 4t)/t$ so $p(t) = -1/t$. The integrating factor is $\mu(t) = e^{\int p(t) dt} = e^{-\ln t} = 1/t$, and multiplying the equation by the integrating factor and then integrating gives us

$$\begin{aligned}\frac{1}{t} \frac{dy}{dt} - \frac{1}{t^2} y &= \frac{1}{t^2} (4t \cos 4t - \sin 4t) \\ \frac{d}{dt} \left(\frac{1}{t} y \right) &= \frac{1}{t^2} (4t \cos 4t - \sin 4t) \\ \frac{1}{t} y &= \frac{\sin 4t}{t} + C \\ y &= \sin 4t + Ct,\end{aligned}$$

where we use the `Integrate` function to evaluate $\int \frac{1}{t^2} (4t \cos 4t - \sin 4t) dt = \frac{\sin 4t}{t} + C$.

$$\text{Integrate}[(4t \cos[4t] - \sin[4t])/t^2, t]$$

$$\frac{\sin[4t]}{t}$$

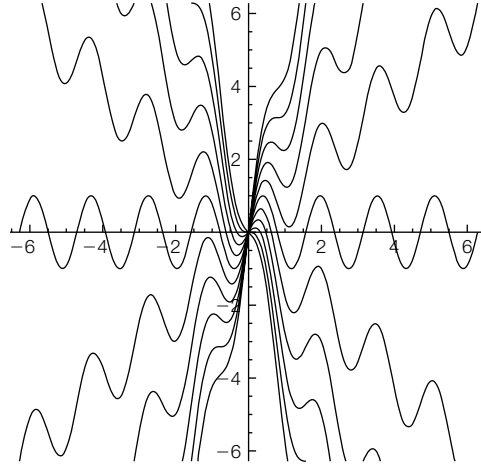
We confirm this result with `DSolve`.

$$\begin{aligned}\text{sol} &= \text{DSolve}[y'[t] - y[t]/t == \\ &\quad (4t \cos[4t] - \sin[4t])/t, y[t], t] \\ &= \{ \{y[t] \rightarrow t C[1] + \sin[4t]\} \}\end{aligned}$$

In the general solution, observe that every solution satisfies $y(0) = 0$. That is, the initial-value problem

$$\frac{dy}{dt} - \frac{1}{t} y = \frac{1}{t^2} (4t \cos 4t - \sin 4t), \quad y(0) = 0$$

has infinitely many solutions. We see this in the plot of several solutions that is generated with `Plot` in Figure 6.6.

**FIGURE 6.6**

Every solution satisfies $y'(0) = 0$

```

toplot = Table[sol/.C[1] → i, {i, -5, 5}];
Plot[y[t]/.toplot, {t, -2Pi, 2Pi}, PlotRange → {-2Pi, 2Pi},
PlotStyle → GrayLevel[0], AspectRatio → Automatic]

```

Application: Free-Falling Bodies

The motion of objects can be determined through the solution of first-order initial-value problems. We begin by explaining some of the theory that is needed to set up the differential equation that models the situation.

Newton's Second Law of Motion: *The rate at which the momentum of a body changes with respect to time is equal to the resultant force acting on the body.*

Because the body's momentum is defined as the product of its mass and velocity, this statement is modeled as

$$\frac{d}{dt}(mv) = F,$$

where m and v represent the body's mass and velocity, respectively, and F is the sum of the forces (the resultant force) acting on the body. Because m is constant, differentiation leads to the well-known equation

$$m \frac{dv}{dt} = F.$$

If the body is subjected only to the force due to gravity, then its velocity is determined by solving the differential equation

$$m \frac{dv}{dt} = mg \quad \text{or} \quad \frac{dv}{dt} = g,$$

where $g = 32\text{ft/s}^2$ (English system) and $g = 9.8\text{m/s}^2$ (metric system). This differential equation is applicable only when the resistive force due to the medium (such as air resistance) is ignored. If this offsetting resistance is considered, we must discuss all of the forces acting on the object. Mathematically, we write the equation as

$$m \frac{dv}{dt} = \sum (\text{forces acting on the object}),$$

where the direction of motion is taken to be the positive direction. Because air resistance acts against the object as it falls and g acts in the same direction of the motion, we state the differential equation in the form

$$m \frac{dv}{dt} = mg + (-F_R) \quad \text{or} \quad m \frac{dv}{dt} = mg - F_R,$$

where F_R represents this resistive force. Note that down is assumed to be the positive direction. The resistive force is typically proportional to the body's velocity, v , or the square of its velocity, v^2 . Hence, the differential equation is linear or nonlinear based on the resistance of the medium taken into account.

Example 6.1.5 An object of mass $m = 1$ is dropped from a height of 50 feet above the surface of a small pond. While the object is in the air, the force due to air resistance is v . However, when the object is in the pond, it is subjected to a buoyancy force equivalent to $6v$. Determine how much time is required for the object to reach a depth of 25 feet in the pond.

Solution This problem must be broken into two parts: an initial-value problem for the object above the pond and an initial-value problem for the object below the surface of the pond. The initial-value problem above the pond's surface is found to be

$$\begin{cases} dv/dt = 32 - v \\ v(0) = 0. \end{cases}$$

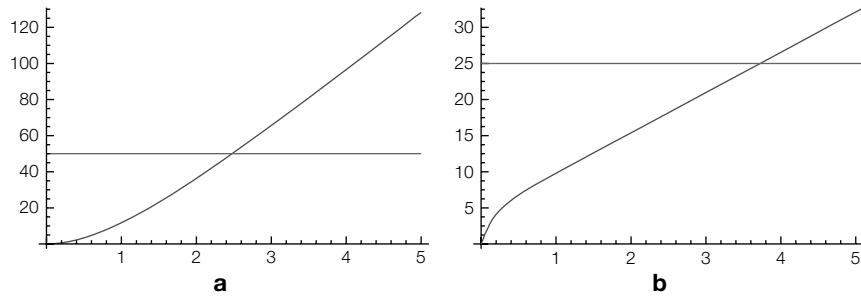
However, to define the initial-value problem to find the velocity of the object beneath the pond's surface, the velocity of the object when it reaches the surface must be known. Hence, the velocity of the object above the surface must be determined by solving the initial-value problem above. The equation $dv/dt = 32 - v$ is separable and solved with DSolve in d1.

Clear[v, y]

d1 = DSolve[{v'[t]==32-v[t], v[0]==0}, v[t], t]

{{v[t] → 32e^{-t} (-1 + e^t)}}

In order to find the velocity when the object hits the pond's surface, we must know the time at which the distance traveled by the object (or the displacement

**FIGURE 6.7**

(a) The object has traveled 50 feet when $t \approx 2.5$. (b) After approximately 4 seconds, the object is 25 feet below the surface of the pond

of the object) is 50. Thus, we must find the displacement function, which is done by integrating the velocity function obtaining $s(t) = 32e^{-t} + 32t - 32$.

```
p1 = DSolve[{y'[t]==v[t]/.d1, y[0]==0}, y[t], t]
{{y[t] -> 32e^-t (1 - e^t + e^t t)}}
```

The displacement function is graphed with **Plot** in Figure 6.7(a). The value of t at which the object has traveled 50 feet is needed. This time appears to be approximately 2.5 seconds.

```
Plot[{y[t]/.p1, 50}, {t, 0, 5}]
```

A more accurate value of the time at which the object hits the surface is found using **FindRoot**. In this case, we obtain $t \approx 2.47864$. The velocity at this time is then determined by substitution into the velocity function resulting in $v(2.47864) \approx 29.3166$. Note that this value is the initial velocity of the object when it hits the surface of the pond.

```
t1 = FindRoot[Evaluate[y[t]/.p1]==50, {t, 2.5}]
{t -> 2.47864}
v1 = d1/.t1
{{v[2.47864] -> 29.3166}}
```

Thus, the initial-value problem that determines the velocity of the object beneath the surface of the pond is given by

$$\begin{cases} dv/dt = 32 - 6v \\ v(0) = 29.3166. \end{cases}$$

The solution of this initial-value problem is $v(t) = \frac{16}{3} + 23.9833e^{-t}$, and integrating to obtain the displacement function (the initial displacement is 0) we obtain $s(t) = 3.99722 - 3.99722e^{-6t} + \frac{16}{3}t$. These steps are carried out in **d2** and **p2**.

```

d2 = DSolve[{v'[t]==32-6v[t], v[0]==v1[[1, 1, 2]]}, v[t], t]
{ {v[t] -> e^{-6t} (23.9832 + 5.33333e^{6t})} }
p2 = DSolve[{y'[t]==v[t]/.d2, y[0]==0}, y[t], t]
{ {y[t] -> e^{-6t} (-3.99721 + 3.99721e^{6t} + 5.33333e^{6t})} }

```

This displacement function is then plotted in Figure 6.7(b) to determine when the object is 25 feet beneath the surface of the pond. This time appears to be near 4 seconds.

```
Plot[{y[t]/.p2, 25}, {t, 0, 5}]
```

A more accurate approximation of the time at which the object is 25 feet beneath the pond's surface is obtained with **FindRoot**. In this case, we obtain $t \approx 3.93802$. Finally, the time required for the object to reach the pond's surface is added to the time needed for it to travel 25 feet beneath the surface to see that approximately 6.41667 seconds are required for the object to travel from a height of 50 feet above the pond to a depth of 25 feet below the surface.

```

t2 = FindRoot[Evaluate[y[t]/.p2]==25, {t, 4}]
{t -> 3.93802}
t1[[1, 2]] + t2[[1, 2]]
6.41667

```

6.1.3 Nonlinear Equations

Mathematica can solve a variety of nonlinear first-order equations that are typically encountered in the introductory differential equations course.

Example 6.1.6 Solve each: (a) $(\cos x + 2xe^y) dx + (\sin y + x^2 e^y - 1) dy = 0$; (b) $(y^2 + 2xy) dx - x^2 dy = 0$.

Solution (a) Notice that $(\cos x + 2xe^y) dx + (\sin y + x^2 e^y - 1) dy = 0$ can be written as $dy/dx = -(\cos x + 2xe^y)/(\sin x + x^2 e^y - 1)$. The equation is an example of an *exact equation*. A theorem tells us that the equation

$$M(x, y)dx + N(x, y)dy = 0$$

is **exact** if and only if $\partial M/\partial y = \partial N/\partial x$.

```

m = Cos[x] + 2x Exp[y];
n = Sin[y] + x^2 Exp[y] - 1;
D[m, y]
D[n, x]
2e^y x
2e^y x

```

We solve exact equations by integrating. Let $F(x, y) = C$ satisfy $(y \cos x + 2xe^y) dx + (\sin y + x^2 e^y - 1) dy = 0$. Then,

$$F(x, y) = \int (\cos x + 2xe^y) dx = \sin x + x^2 e^y + g(y),$$

where $g(y)$ is a function of y .

```
f1 = Integrate[m, x]  
eyx2 + Sin[x]
```

We next find that $g'(y) = \sin y - 1$ so $g(y) = -\cos y - y$. Hence, a general solution of the equation is

$$\sin x + x^2 e^y - \cos y - y = C.$$

```
f2 = D[f1, y]  
eyx2  
f3 = Solve[f2 + c == n, c]  
{c → -1 + Sin[y]}
```

```
Integrate[f3[[1, 1, 2]], y]  
-y - Cos[y]
```

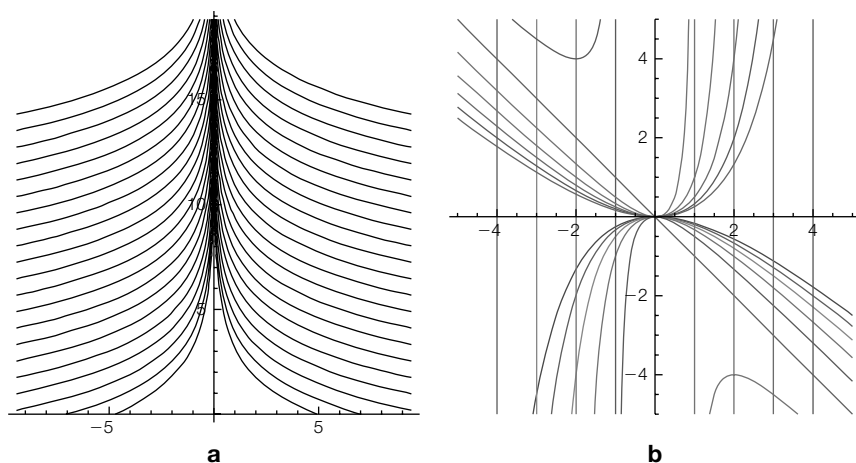
We confirm this result with `DSolve`. Notice that Mathematica warns us that it cannot solve for y explicitly and returns the same implicit solution obtained by us.

```
mf = m/.y → y[x];  
nf = n/.y → y[x];  
sol = DSolve[mf + nf y'[x] == 0, y[x], x]  
Solve::tdep : The equations appear to involve the variables to be solved for in  
an essentially non-algebraic way.))  
Solve [ey[x]x2 - Cos[y[x]] + Sin[x] - y[x] == C[1], y[x]]
```

Graphs of several solutions using the values of C generated in `cvals` are graphed with `ContourPlot` in Figure 6.8.

```
sol2 = sol[[1, 1]]/.y[x] → y  
eyx2 - y - Cos[y] + Sin[x]  
cvals = Table[sol2/.{x → -3Pi/2, y → i}, {i, 0, 6Pi, 6Pi/24}];  
ContourPlot[sol2, {x, -3Pi, 3Pi}, {y, 0, 6Pi}, Contours → cvals,  
ContourShading → False, Axes → Automatic, Frame → False,  
AxesOrigin → {0, 0}, ContourStyle → GrayLevel[0]]
```

(b) We can write $(y^2 + 2xy) dx - x^2 dy = 0$ as $dy/dx = (y^2 + 2xy)/x^2$. A first-order equation is **homogeneous** if it can be written in the form $dy/dx = F(y/x)$. Homogeneous equations are reduced to separable equations with either the substitution $y = ux$ or $x = vy$. In this case, we have that $dy/dx = (y/x)^2 + 2(y/x)$, so the equation is homogeneous.

**FIGURE 6.8**

(a) Graphs of several solutions of $(\cos x + 2xe^y) dx + (\sin y + x^2 e^y - 1) dy = 0$. (b) Graphs of several solutions of $(y^2 + 2xy) dx - x^2 dy = 0$

Let $y = ux$. Then, $dy = u dx + x du$. Substituting into $(y^2 + 2xy) dx - x^2 dy = 0$ and separating gives us

$$\begin{aligned}
 (y^2 + 2xy) dx - x^2 dy &= 0 \\
 (u^2 x^2 + 2ux^2) dx - x^2(u dx + x du) &= 0 \\
 (u^2 + 2u) dx - (u dx + x du) &= 0 \\
 (u^2 + u) dx &= x du \\
 \frac{1}{u(u+1)} du &= \frac{1}{x} dx.
 \end{aligned}$$

Integrating the left- and right-hand sides of this equation with **Integrate**,

```

Integrate[1/(u(u + 1)), u]
Log[u] - Log[1 + u]
Integrate[1/x, x]
Log[x]

```

exponentiating, resubstituting $u = y/x$, and solving for y gives us

$$\begin{aligned}
 \ln |u| - \ln |u + 1| &= \ln |x| + C \\
 \frac{u}{u + 1} &= Cx \\
 \frac{y}{\frac{y}{x} + 1} &= Cx \\
 y &= \frac{Cx^2}{1 - Cx}.
 \end{aligned}$$

```
sol1 = Solve[(y/x)/(y/x + 1) == cx, y]

$$\left\{ \left\{ y \rightarrow -\frac{cx^2}{-1 + cx} \right\} \right\}$$

```

We confirm this result with DSolve and then graph several solutions with Plot in Figure 6.8(b).

```
sol2 = DSolve[y[x]^2 + 2x y[x] - x^2 y'[x] == 0, y[x], x]

$$\left\{ \left\{ y[x] \rightarrow -\frac{x^2}{x - C[1]} \right\} \right\}$$

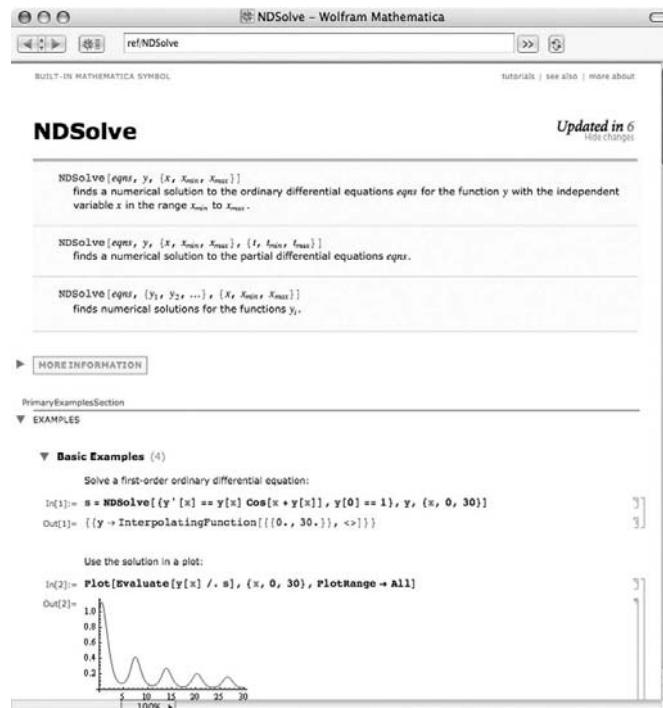
toplot = Table[sol2[[1, 1, 2]]/.C[1] -> i, {i, -5, 5}];
Plot[Tooltip[toplot], {x, -5, 5}, PlotRange -> {-5, 5},
AspectRatio -> Automatic]
```

6.1.4 Numerical Methods

If numerical results are desired, use NDSolve:

```
NDSolve[{y'[t] == f[t, y[t]], y[t0] == y0}, y[t], {t, a, b}]
```

attempts to generate a numerical solution of $dy/dt = f(t, y)$, $y(t_0) = y_0$, valid for $a \leq t \leq b$.



Example 6.1.7 Consider $dy/dt = (t^2 - y^2) \sin y$, $y(0) = -1$. (a) Determine $y(1)$. (b) Graph $y(t)$, $-1 \leq t \leq 10$.

Solution We first remark that DSolve can neither exactly solve the differential equation $y' = (t^2 - y^2) \sin y$ nor find the solution that satisfies $y(0) = -1$.

```
sol = DSolve[y'[t] == (t^2 - y[t]^2) Sin[y[t]], y[t], t]
Solve::tdep: The equations appear to involve the variables to be solved for in an essentially non-algebraic way. >>
DSolve[y'[t] == Sin[y[t]] (t^2 - y[t]^2), y[t], t]

sol = DSolve[{y'[t] == (t^2 - y[t]^2) Sin[y[t]], y[0] == -1}, y[t], t]
Solve::tdep: The equations appear to involve the variables to be solved for in an essentially non-algebraic way. >>
DSolve[{y'[t] == Sin[y[t]] (t^2 - y[t]^2), y[0] == -1}, y[t], t]
```

However, we obtain a numerical solution valid for $0 \leq t \leq 1000$ using the NDSolve function.

```
sol = NDSolve[{y'[t] == (t^2 - y[t]^2) Sin[y[t]], y[0] == -1}, y[t],
{t, -1, 10}]
{{y[t] -> InterpolatingFunction[{{-1., 10.}}, <>][t]}}
```

Entering `sol /. t -> 1` evaluates the numerical solution if $t = 1$.

```
sol /. t -> 1
{{y[1] -> -0.766013}}
```

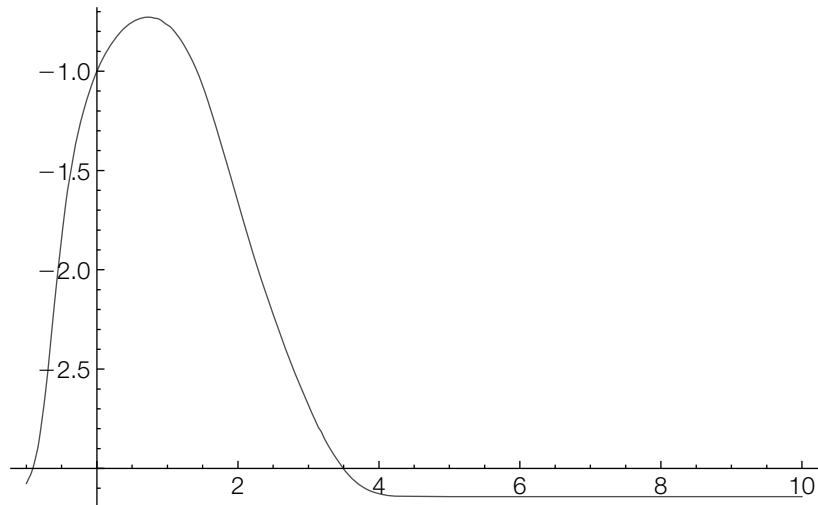
The result means that $y(1) \approx -0.766$. We use the Plot command to graph the solution for $0 \leq t \leq 10$ in Figure 6.9.

```
Plot[y[t] /. sol, {t, -1, 10}]
```

Example 6.1.8 (Logistic Equation with Predation). Incorporating predation into the **logistic equation**, $y' = \alpha y \left(1 - \frac{1}{K} y\right)$, results in $\frac{dy}{dt} = \alpha y \left(1 - \frac{1}{K} y\right) - P(y)$, where $P(y)$ is a function of y describing the rate of predation. A typical choice for P is $P(y) = ay^2/(b^2 + y^2)$ because $P(0) = 0$ and P is bounded above: $\lim_{t \rightarrow \infty} P(y) < \infty$.

Remark 6.2 Of course, if $\lim_{t \rightarrow \infty} y(t) = Y$, then $\lim_{t \rightarrow \infty} P(y) = aY^2/(b^2 + Y^2)$. Generally, however, $\lim_{t \rightarrow \infty} P(y) \neq a$ because $\lim_{t \rightarrow \infty} y(t) \leq K \neq \infty$, for some $K \geq 0$, in the predation situation.

If $\alpha = 1$, $a = 5$, and $b = 2$, graph the direction field associated with the equation as well as various solutions if (a) $K = 19$ and (b) $K = 20$.

**FIGURE 6.9**

Graph of the solution to $y' = (t^2 - y^2) \sin y$, $y(0) = -1$

Solution (a) We define `eqn[k]` to be $\frac{dy}{dt} = y \left(1 - \frac{1}{K} y \right) - \frac{5y^2}{4 + y^2}$.

Needs["VectorFieldPlots"]

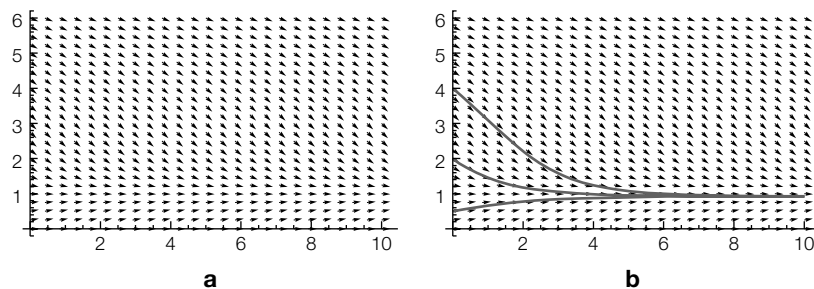
eqn[k_] = y'[t] == y[t](1 - y[t]/k) - 5y[t]^2/(4 + y[t]^2);

We use `VectorPlotField` to graph the direction field in Figure 6.10(a) and then the direction field along with the solutions that satisfy $y(0) = 0.5$, $y(0) = 0.2$, and $y(0) = 4$ in Figure 6.10(b).

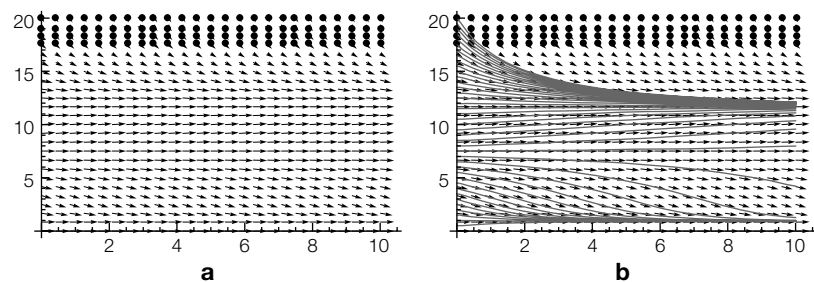
```
pvf19 = Show[VectorFieldPlot[{1, y(1 - 1/19y) - 5y^2/(4 + y^2)}, {t, 0, 10},
  {y, 0, 6}, ScaleFunction -> (1&), PlotPoints -> 25],
  Axes -> Automatic, AxesOrigin -> {0, 0}];
numsols = Map[NDSolve[{eqn[19], y[0] == #}, y[t], {t, 0, 10}]&,
  {0.5, 2, 4}];
solplot = Plot[y[t]/.numsols, {t, 0, 10}, PlotRange -> All,
  PlotStyle -> {{GrayLevel[.4], Thickness[.01]}}];
Show[GraphicsRow[{pvf19, Show[pvf19, solplot]}]]
```

In the plot, notice that all nontrivial solutions appear to approach an equilibrium solution. We determine the equilibrium solution by solving $y' = 0$

```
eqn[19][[2]]
(1 - y[t]/19) y[t] - 5y[t]^2/(4 + y[t]^2)
```

**FIGURE 6.10**

(a) Direction field and (b) direction field with three solutions

**FIGURE 6.11**

(a) Direction field and (b) direction field with several solutions

```
Solve[eqn[19.][[2]]==0, y[t]]
{{y[t]-> 0.}, {y[t]-> 0.923351}, {y[t]-> 9.03832 - 0.785875 I},
 {y[t]-> 9.03832 + 0.785875 I}}
```

to see that it is $y \approx 0.923$.

(b) We carry out similar steps for (b). First, we graph the direction field with `PlotVectorField` in Figure 6.11(a).

```
pvf20 = Show[VectorFieldPlot[{1, y(1 - 1/20y) - 5y^2/(4 + y^2)}, {t, 0, 10},
 {y, 0, 20}, ScaleFunction -> (1&), PlotPoints -> 25],
 Axes -> Automatic, AxesOrigin -> {0, 0},
 AspectRatio -> 1/GoldenRatio];
```

We then use `Map` together with `NDSolve` to numerically find the solution satisfying $y(0) = 0.5i$, for $i = 1, 2, \dots, 40$ and name the resulting list `numsols`. The functions contained in `numsols` are graphed with `Plot` in `solplot`.

```
numsols = Map[NDSolve[{eqn[20], y[0]==#}, y[t], {t, 0, 10}]&,
 Table[0.5i, {i, 1, 40}]];
solplot = Plot[y[t]/.numsols, {t, 0, 10}, PlotRange -> All,
 PlotStyle -> {{GrayLevel[.4], Thickness[.005]}}];
```

Last, we display the direction field along with the solution graphs in `solplot` using `Show` in Figure 6.11(b).

Show[GraphicsRow[{pvf20, Show[pvf20, solplot]}]]

Notice that there are three nontrivial equilibrium solutions that are found by solving $y' = 0$.

Solve[eqn[20.][[2]]==0, y[t], t]

$\{\{y[t] \rightarrow 0.\}, \{y[t] \rightarrow 0.926741\}, \{y[t] \rightarrow 7.38645\}, \{y[t] \rightarrow 11.6868\}\}$

In this case, $y \approx 0.926$ and $y \approx 11.687$ are stable, whereas $y \approx 7.386$ is unstable.

6.2 SECOND-ORDER LINEAR EQUATIONS

We now present a concise discussion of second-order linear equations, which are extensively discussed in the introductory differential equations course.

6.2.1 Basic Theory

The **general form** of the **second-order linear equation** is

$$a_2(t) \frac{d^2 y}{dt^2} + a_1(t) \frac{dy}{dt} + a_0(t) y = f(t), \quad (6.7)$$

where $a_2(t)$ is not identically the zero function.

The **standard form** of the second-order linear equation (6.7) is

$$\frac{d^2 y}{dt^2} + p(t) \frac{dy}{dt} + q(t) y = f(t). \quad (6.8)$$

The **corresponding homogeneous equation** of equation (6.8) is

$$\frac{d^2 y}{dt^2} + p(t) \frac{dy}{dt} + q(t) y = 0. \quad (6.9)$$

A **general solution** of equation (6.9) is $y = c_1 y_1 + c_2 y_2$, where

1. y_1 and y_2 are solutions of equation (6.9), and
2. y_1 and y_2 are *linearly independent*.

If y_1 and y_2 are solutions of equation (6.9), then y_1 and y_2 are **linearly independent** if and only if the **Wronskian**,

$$W(\{y_1, y_2\}) = \begin{vmatrix} y_1 & y_2 \\ y_1' & y_2' \end{vmatrix} = y_1 y_2' - y_1' y_2, \quad (6.10)$$

is not the zero function. If y_1 and y_2 are linearly independent solutions of equation (6.9), we call the set $S = \{y_1, y_2\}$ a **fundamental set of solutions** for equation (6.9).

A particular solution, y_p , is a solution that does not contain any arbitrary constants.

Let y be a general solution of equation (6.8) and y_p be a particular solution of equation (6.8). It follows that $y - y_p$ is a solution of equation (6.9), so $y - y_p = y_b$, where y_b is a general solution of equation (6.9). Hence, $y = y_b + y_p$. That is, to solve the nonhomogeneous equation, we need a general solution, y_b , of the corresponding homogeneous equation and a particular solution, y_p , of the nonhomogeneous equation.

6.2.2 Constant Coefficients

Suppose that the coefficient functions of equation (6.7) are constants: $a_2(t) = a$, $a_1(t) = b$, and $a_0(t) = c$, and that $f(t)$ is identically the zero function. In this case, equation (6.7) becomes

$$ay'' + by' + cy = 0. \quad (6.11)$$

Now suppose that $y = e^{kt}$, k constant, is a solution of equation (6.11). Then, $y' = ke^{kt}$ and $y'' = k^2e^{kt}$. Substitution into equation (6.11) then gives us

$$\begin{aligned} ay'' + by' + cy &= ak^2e^{kt} + bke^{kt} + ce^{kt} \\ &= e^{kt}(ak^2 + bk + c) = 0. \end{aligned}$$

Because $e^{kt} \neq 0$, the solutions of equation (6.11) are determined by the solutions of

$$ak^2 + bk + c = 0, \quad (6.12)$$

called the **characteristic equation** of equation (6.11).

Theorem 1. *Let k_1 and k_2 be the solutions of equation (6.12).*

1. *If $k_1 \neq k_2$ are real and distinct, two linearly independent solutions of equation (6.11) are $y_1 = e^{k_1t}$ and $y_2 = e^{k_2t}$; a general solution of equation (6.11) is*

$$y = c_1e^{k_1t} + c_2e^{k_2t}.$$

2. *If $k_1 = k_2$, two linearly independent solutions of equation (6.11) are $y_1 = e^{k_1t}$ and $y_2 = te^{k_1t}$; a general solution of equation (6.11) is*

$$y = c_1e^{k_1t} + c_2te^{k_1t}.$$

3. *If $k_{1,2} = \alpha \pm \beta i$, $\beta \neq 0$, two linearly independent solutions of equation (6.11) are $y_1 = e^{\alpha t} \cos \beta t$ and $y_2 = e^{\alpha t} \sin \beta t$; a general solution of equation (6.11) is*

$$y = e^{\alpha t} (c_1 \cos \beta t + c_2 \sin \beta t).$$

Example 6.2.1 Solve each of the following equations: (a) $6y'' + y' - 2y = 0$; (b) $y'' + 2y' + y = 0$; (c) $16y'' + 8y' + 145y = 0$.

Solution (a) The characteristic equation is $6k^2 + k - 2 = (3k + 2)(2k - 1) = 0$ with solutions $k = -2/3$ and $k = 1/2$. We check with either **Factor** or **Solve**.

```
Factor[6k^2 + k - 2]
(-1 + 2k)(2 + 3k)
Solve[6k^2 + k - 2 == 0]
{{k -> -2/3}, {k -> 1/2}}
```

Then, a fundamental set of solutions is $\{e^{-2t/3}, e^{t/2}\}$ and a general solution is

$$y = c_1 e^{-2t/3} + c_2 e^{t/2}.$$

Of course, we obtain the same result with **DSolve**.

```
Clear[y]
DSolve[6y''[t] + y'[t] - 2y[t] == 0, y[t], t]
{{y[t] -> e^{-2t/3} C[1] + e^{t/2} C[2]}}
```

(b) The characteristic equation is $k^2 + 2k + 1 = (k + 1)^2 = 0$ with solution $k = -1$, which has multiplicity two, so a fundamental set of solutions is $\{e^{-t}, te^{-t}\}$ and a general solution is

$$y = c_1 e^{-t} + c_2 t e^{-t}.$$

We check the calculation in the same way as in (a).

```
Factor[k^2 + 2k + 1]
Solve[k^2 + 2k + 1 == 0]
DSolve[y''[t] + 2y'[t] + y[t] == 0, y[t], t]
(1 + k)^2
{{k -> -1}, {k -> -1}}
{{y[t] -> e^{-t} C[1] + e^{-t} t C[2]}}
```

(c) The characteristic equation is $16k^2 + 8k + 145 = 0$ with solutions $k_{1,2} = -\frac{1}{4} \pm 3i$, so a fundamental set of solutions is $\{e^{-t/4} \cos 3t, e^{-t/4} \sin 3t\}$ and a general solution is

$$y = e^{-t/4} (c_1 \cos 3t + c_2 \sin 3t).$$

The calculation is verified in the same way as in (a) and (b).

Factor[16k^2 + 8k + 145, GaussianIntegers → True]

$((1 - 12i) + 4k)((1 + 12i) + 4k)$

Solve[16k^2 + 8k + 145 == 0]

$\left\{ \left\{ k \rightarrow -\frac{1}{4} - 3i \right\}, \left\{ k \rightarrow -\frac{1}{4} + 3i \right\} \right\}$

DSolve[16y''[t] + 8y'[t] + 145y[t] == 0, y[t], t]

$\left\{ \left\{ y[t] \rightarrow e^{-t/4} C[2] \cos[3t] + e^{-t/4} C[1] \sin[3t] \right\} \right\}$

Example 6.2.2 Solve $64 \frac{d^2 y}{dt^2} + 16 \frac{dy}{dt} + 1025y = 0$, $y(0) = 1$, $\frac{dy}{dt}(0) = 3$.

Solution A general solution of $64y'' + 16y' + 1025y = 0$ is $y = e^{-t/8} (c_1 \sin 4t + c_2 \cos 4t)$.

gensol = DSolve[64y''[t] + 16y'[t] + 1025y[t] == 0, y[t], t]

$\left\{ \left\{ y[t] \rightarrow e^{-t/8} C[2] \cos[4t] + e^{-t/8} C[1] \sin[4t] \right\} \right\}$

Applying $y(0) = 1$ shows us that $c_2 = 1$.

e1 = gensol[[1, 1, 2]]/.t → 0

$C[2]$

Computing y'

D[y[t]/.gensol[[1]], t]

$4e^{-t/8} C[1] \cos[4t] - \frac{1}{8} e^{-t/8} C[2] \cos[4t] - \frac{1}{8} e^{-t/8} C[1] \sin[4t] - 4e^{-t/8} C[2] \sin[4t]$

and then $y'(0)$, shows us that $-4c_1 - \frac{1}{8}c_2 = 3$.

e2 = D[y[t]/.gensol[[1]], t]/.t → 0

$4C[1] - \frac{C[2]}{8}$

Solving for c_1 and c_2 with Solve shows us that $c_1 = -25/32$ and $c_2 = 1$.

cvals = Solve[{e1 == 1, e2 == 3}]

$\left\{ \left\{ C[1] \rightarrow \frac{25}{32}, C[2] \rightarrow 1 \right\} \right\}$

Thus, $y = e^{-t/8} \left(\frac{-25}{32} \sin 4t + \cos 4t \right)$, which we graph with Plot in Figure 6.12.

sol = y[t]/.gensol[[1]]/.cvals[[1]]

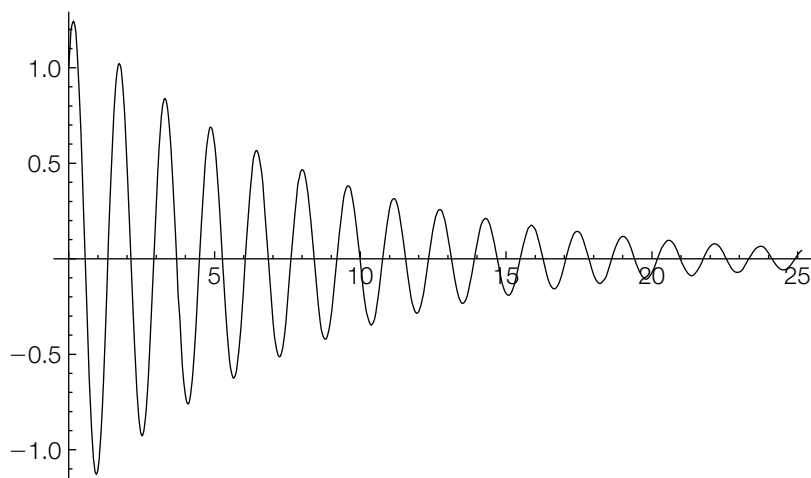
$e^{-t/8} \cos[4t] + \frac{25}{32} e^{-t/8} \sin[4t]$

Plot[sol, {t, 0, 8Pi}]

We verify the calculation with DSolve.

DSolve[{64y''[t] + 16y'[t] + 1025y[t] == 0, y[0] == 1, y'[0] == 3}, y[t], t]

$\left\{ \left\{ y[t] \rightarrow \frac{1}{32} e^{-t/8} (32 \cos[4t] + 25 \sin[4t]) \right\} \right\}$

**FIGURE 6.12**

The solution to the initial-value problem tends to 0 as $t \rightarrow \infty$

Application: Harmonic Motion

Suppose that a mass is attached to an elastic spring that is suspended from a rigid support such as a ceiling. According to Hooke's law, the spring exerts a restoring force in the upward direction that is proportional to the displacement of the spring.

Hooke's Law: $F = ks$, where $k > 0$ is the constant of proportionality or spring constant, and s is the displacement of the spring.

Using Hooke's law and assuming that $x(t)$ represents the displacement of the mass from the equilibrium position at time t , we obtain the initial-value problem

$$m \frac{d^2 x}{dt^2} + kx = 0, \quad x(0) = \alpha, \quad \frac{dx}{dt}(0) = \beta.$$

Note that the initial conditions give the initial displacement and velocity, respectively. This differential equation disregards all retarding forces acting on the motion of the mass and a more realistic model that takes these forces into account is needed. Studies in mechanics reveal that resistive forces due to damping are proportional to a power of the velocity of the motion. Hence, $F_R = a \, dx/dt$ or $F_R = a \, (dx/dt)^3$, where $a > 0$, are typically used to represent the damping force. Then, we have the following initial-value problem assuming that $F_R = a \, dx/dt$:

$$m \frac{d^2 x}{dt^2} + a \frac{dx}{dt} + kx = 0, \quad x(0) = \alpha, \quad \frac{dx}{dt}(0) = \beta.$$

Problems of this type are characterized by the value of $a^2 - 4mk$ as follows:

1. $a^2 - 4mk > 0$. This situation is said to be **overdamped** because the damping coefficient a is large in comparison to the spring constant k .
2. $a^2 - 4mk = 0$. This situation is described as **critically damped** because the resulting motion is oscillatory with a slight decrease in the damping coefficient a .
3. $a^2 - 4mk < 0$. This situation is called **underdamped** because the damping coefficient a is small in comparison with the spring constant k .

Example 6.2.3 Classify the following differential equations as overdamped, underdamped, or critically damped. Also, solve the corresponding initial-value problem using the given initial conditions and investigate the behavior of the solutions.

- (a) $\frac{d^2x}{dt^2} + 8\frac{dx}{dt} + 16x = 0$ subject to $x(0) = 0$ and $\frac{dx}{dt}(0) = 1$;
- (b) $\frac{d^2x}{dt^2} + 5\frac{dx}{dt} + 4x = 0$ subject to $x(0) = 1$ and $\frac{dx}{dt}(0) = 1$; and
- (c) $\frac{d^2x}{dt^2} + \frac{dx}{dt} + 16x = 0$ subject to $x(0) = 0$ and $\frac{dx}{dt}(0) = 1$.

Solution For (a), we identify $m = 1$, $a = 8$, and $k = 16$ so that $a^2 - 4mk = 0$, which means that the differential equation $x'' + 8x' + 16x = 0$ is critically damped. After defining `de1`, we solve the equation subject to the initial conditions and name the resulting output `sol1`. We then graph the solution shown in Figure 6.13(a).

```
Clear[de1, x, t]
de1 = x''[t] + 8x'[t] + 16x[t] == 0;
sol1 = DSolve[{de1, x[0] == 0, x'[0] == 1}, x[t], t]
{ {x[t] -> e^{-4t}t} }
p1 = Plot[sol1][[1, 1, 2]], {t, 0, 4}]
```

For (b), we proceed in the same manner. We identify $m = 1$, $a = 5$, and $k = 4$ so that $a^2 - 4mk = 9$ and the equation $x'' + 5x' + 4x = 0$ is overdamped. We then

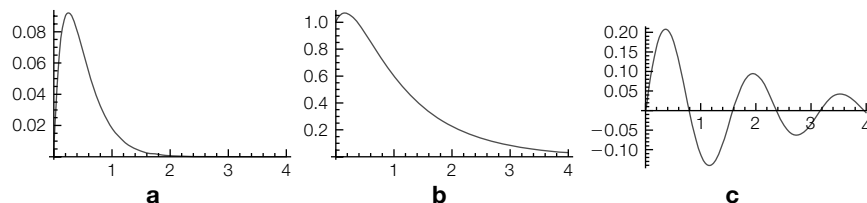


FIGURE 6.13

(a) Critically damped motion. (b) Overdamped motion. (c) Underdamped motion

define `de2` to be the equation and the solution to the initial-value problem obtained with `DSolve`, `sol2`, and then graph $x(t)$ on the interval $[0, 4]$ in Figure 6.13(b).

```
Clear[de2, x, t]
de2 = x''[t] + 5x'[t] + 4x[t]==0;
sol2 = DSolve[{de2, x[0]==1, x'[0]==1}, x[t], t]
{{x[t] -> 1/3 e^{-4t} (-2 + 5e^{3t})}}
p2 = Plot[sol2[[1, 1, 2]], {t, 0, 4}]
```

For (c), we proceed in the same manner as in (a) and (b) to show that the equation is underdamped because the value of $a^2 - 4mk$ is -63 . See Figure 6.13(c).

You can also use `Manipulate` to help you visualize harmonic motion. With

```
Manipulate[
  sol = DSolve[{mx''[t] + ax'[t] + kx[t]==0, x[0]==0, x'[0]==1}, x[t], t];
  Plot[x[t]/.sol, {t, 0, 5}, PlotRange -> {-1/2, 1/2}, AspectRatio -> 1,
    {{m, 1}, 0, 5}, {{a, 8}, 0, 15, 1}, {{k, 16}, 0, 20, 1}]
```

we generate a `Manipulate` object that lets us investigate harmonic motion for various values of m , a , and k if the initial position is zero ($x(0) = 0$) and the initial velocity is one ($x'(0) = 1$). See Figure 6.14. (Note that m is centered at 1, a at 8, and k at 16.)

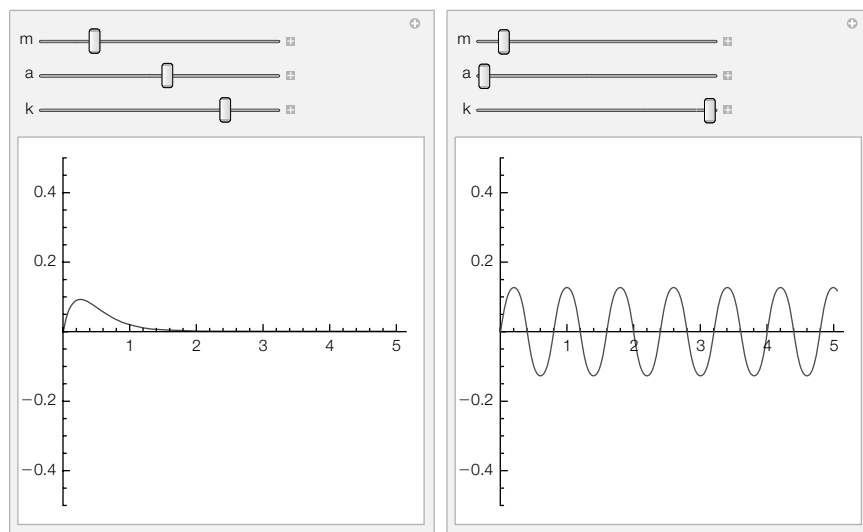


FIGURE 6.14

Using `Manipulate` to investigate harmonic motion

6.2.3 Undetermined Coefficients

If equation (6.7) has constant coefficients and $f(t)$ is a product of terms t^n , $e^{\alpha t}$, α constant, $\cos \beta t$, and/or $\sin \beta t$, β constant, *undetermined coefficients* can often be used to find a particular solution of equation (6.7). The key to implementing the method is to *judiciously* choose the correct form of y_p .

Assume that a general solution, y_h , of the corresponding homogeneous equation has been found and that each term of $f(t)$ has the form

$$t^n e^{\alpha t} \cos \beta t \quad \text{or} \quad t^n e^{\alpha t} \sin \beta t.$$

For *each* term of $f(t)$, write down the *associated set*

$$F = \{t^n e^{\alpha t} \cos \beta t, t^n e^{\alpha t} \sin \beta t, t^{n-1} e^{\alpha t} \cos \beta t, t^{n-1} e^{\alpha t} \sin \beta t, \dots, e^{\alpha t} \cos \beta t, e^{\alpha t} \sin \beta t\}.$$

If any element of F is a solution to the corresponding homogeneous equation, multiply each element of F by t^m , where m is the smallest positive integer so that none of the elements of $t^m F$ are solutions to the corresponding homogeneous equation. A particular solution will be a linear combination of the functions in all the F 's.

Example 6.2.4 Solve $4 \frac{d^2 y}{dt^2} - y = t - 2 - 5 \cos t - e^{-t/2}$.

Solution The corresponding homogeneous equation is $4y'' - y = 0$ with general solution $y_h = c_1 e^{-t/2} + c_2 e^{t/2}$.

DSolve[4y''[t]-y[t]==0, y[t], t]
 $\{ \{y[t] \rightarrow e^{t/2} C[1] + e^{-t/2} C[2]\} \}$

A fundamental set of solutions for the corresponding homogeneous equation is $S = \{e^{-t/2}, e^{t/2}\}$. The associated set of functions for $t - 2$ is $F_1 = \{1, t\}$, the associated set of functions for $-5 \cos t$ is $F_2 = \{\cos t, \sin t\}$, and the associated set of functions for $-e^{-t/2}$ is $F_3 = \{e^{-t/2}\}$. Note that $e^{-t/2}$ is an element of S so we multiply F_3 by t resulting in $tF_3 = \{te^{-t/2}\}$. Then, we search for a particular solution of the form

$$y_p = A + Bt + C \cos t + D \sin t + Ete^{-t/2},$$

No element of F_1 is contained in S and no element of F_2 is contained in S .

We do not use capital letters so as to avoid any confusion with built-in Mathematica commands.

where A , B , C , D , and E are constants to be determined.

yp[t_] = a + bt + c Cos[t] + d Sin[t] + et Exp[-t/2]
 $a + bt + ee^{-t/2} + c \cos[t] + d \sin[t]$

Computing y'_p and y''_p

dyp = yp'[t]
 $b + ee^{-t/2} - \frac{1}{2}ee^{-t/2}t + d \cos[t] - c \sin[t]$
d2yp = yp''[t]
 $-ee^{-t/2} + \frac{1}{4}ee^{-t/2}t - c \cos[t] - d \sin[t]$

and substituting into the nonhomogeneous equation results in

$$-A - Bt - 5C \cos t - 5D \sin t - 4Ee^{-t/2} = t - 2 - 5 \cos t - e^{-t/2}.$$

$$\text{eqn} = 4y''[t] - y[t] == t - 2 - 5\cos[t] - \text{Exp}[-t/2]$$

$$-a - bt - ee^{-t/2} - c\cos[t] - d\sin[t] + 4(-ee^{-t/2} + \frac{1}{4}ee^{-t/2}t - c\cos[t] - d\sin[t]) == -2 - e^{-t/2} + t - 5\cos[t]$$

Equating coefficients results in

$$-A = -2 \quad -B = 1 \quad -5C = -5 \quad -5D = 0 \quad -4E = -1$$

so $A = 2$, $B = -1$, $C = 1$, $D = 0$, and $E = 1/4$.

$$\text{cvals} = \text{Solve}[\{-a == -2, -b == 1, -5c == -5, -5d == 0, -4e == -1\}]$$

$$\{\{a \rightarrow 2, b \rightarrow -1, c \rightarrow 1, d \rightarrow 0, e \rightarrow \frac{1}{4}\}\}$$

y_p is then given by $y_p = 2 - t + \cos t + \frac{1}{4}te^{-t/2}$

$$yp[t]/.\text{cvals}[[1]]$$

$$2 - t + \frac{1}{4}e^{-t/2}t + \cos[t]$$

and a general solution is given by

$$y = y_b + y_p = c_1 e^{-t/2} + c_2 e^{t/2} + 2 - t + \cos t + \frac{1}{4}te^{-t/2}.$$

Note that $-A - Bt - 5C \cos t - 5D \sin t - 4Ee^{-t/2} = t - 2 - 5 \cos t - e^{-t/2}$ is true for *all* values of t . Evaluating for five different values of t gives us five equations that we then solve for A , B , C , D , and E , resulting in the same solutions as already obtained.

$$\text{e1} = \text{eqn}/.t \rightarrow 0$$

$$-a - c + 4(-c - e) == -8$$

$$\text{e2} = \text{eqn}/.t \rightarrow 1;$$

$$\text{e3} = \text{eqn}/.t \rightarrow 2;$$

$$\text{e4} = \text{eqn}/.t \rightarrow 3;$$

$$\text{e5} = \text{eqn}/.t \rightarrow 4;$$

$$\text{Solve}[\{\text{e1}, \text{e2}, \text{e3}, \text{e4}, \text{e5}\}]/\text{Simplify}$$

$$\{\{b \rightarrow -1, d \rightarrow 0, a \rightarrow 2, c \rightarrow 1, e \rightarrow \frac{1}{4}\}\}$$

Last, we check our calculation with DSolve and Simplify.

$$\text{DSolve}[4y''[t] - y[t] == t - 2 - 5\cos[t] - \text{Exp}[-t/2], y[t], t]/\text{Simplify}$$

$$\{\{y[t] \rightarrow \frac{1}{4}e^{-t/2}(1 - 4e^{t/2}(-2 + t) + t + 4e^t C[1] + 4C[2] + 4e^{t/2}\cos[t])\}\}$$

Example 6.2.5 Solve $y'' + 4y = \cos 2t$, $y(0) = 0$, $y'(0) = 0$.

Solution A general solution of the corresponding homogeneous equation is $y_h = c_1 \cos 2t + c_2 \sin 2t$. For this equation, $F = \{\cos 2t, \sin 2t\}$. Because elements of F are solutions to the corresponding homogeneous equation, we multiply each element of F by t resulting in $tF = \{t \cos 2t, t \sin 2t\}$. Therefore, we assume that a particular solution has the form

$$y_p = At \cos 2t + Bt \sin 2t,$$

where A and B are constants to be determined. Proceeding in the same manner as before, we compute y'_p and y''_p

$$\begin{aligned} y_p[t] &= a t \cos[2t] + b t \sin[2t] \\ y'_p[t] & \\ y''_p[t] & \\ a t \cos[2t] + b t \sin[2t] & \\ a \cos[2t] + 2b t \cos[2t] + b \sin[2t] - 2a t \sin[2t] & \\ 4b \cos[2t] - 4a t \cos[2t] - 4a \sin[2t] - 4bt \sin[2t] & \end{aligned}$$

and then substitute into the nonhomogeneous equation.

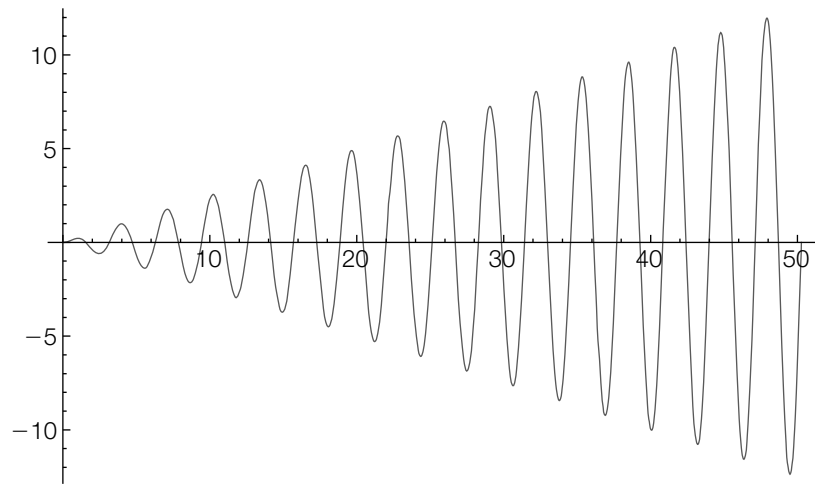
$$\begin{aligned} \text{eqn} &= y''_p[t] + 4y_p[t] == \cos[2t] \\ 4b \cos[2t] - 4a t \cos[2t] - 4a \sin[2t] - 4b t \sin[2t] + 4(a t \cos[2t] + b t \sin[2t]) & \\ == \cos[2t] & \end{aligned}$$

Equating coefficients readily yields $A = 0$ and $B = 1/4$. Alternatively, remember that $-4A \sin 2t + 4B \cos 2t = \cos 2t$ is true for *all* values of t . Evaluating for two values of t and then solving for A and B gives the same result.

$$\begin{aligned} e1 &= \text{eqn}/t \rightarrow 0 \\ e2 &= \text{eqn}/t \rightarrow 1 \\ \text{cvals} &= \text{Solve}[\{e1, e2\}] \\ 4b &== 1 \\ -4a \cos[2] + 4b \cos[2] - 4a \sin[2] - 4b \sin[2] + 4(a \cos[2] + b \sin[2]) &== \cos[2] \\ \{ \{a \rightarrow 0, b \rightarrow \frac{1}{4}\} \} & \end{aligned}$$

It follows that $y_p = \frac{1}{4}t \sin 2t$ and $y = c_1 \cos 2t + c_2 \sin 2t + \frac{1}{4}t \sin 2t$.

$$\begin{aligned} y_p[t]/\text{cvals}[[1]] & \\ \frac{1}{4}t \sin[2t] & \\ y[t] &= c1 \cos[2t] + c2 \sin[2t] + 1/4 t \sin[2t] \\ c1 \cos[2t] + c2 \sin[2t] + \frac{1}{4}t \sin[2t] & \end{aligned}$$

**FIGURE 6.15**

The forcing function causes the solution to become unbounded as $t \rightarrow \infty$

Applying the initial conditions after finding y'

```
y'[t]
2c2 Cos[2t] + 1/2 t Cos[2t] + 1/4 Sin[2t] - 2c1 Sin[2t]
cvals = Solve[{y[0]==0, y'[0]==0}]
{{c1 -> 0, c2 -> 0}}
```

results in $y = \frac{1}{4}t \sin 2t$, which we graph with Plot in Figure 6.15.

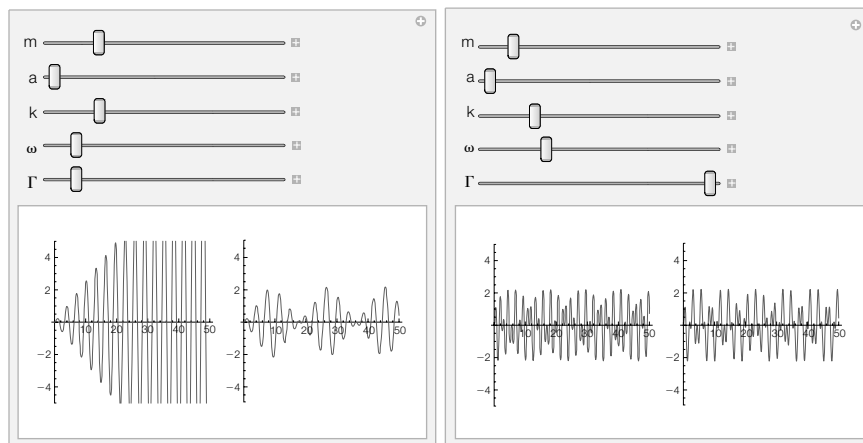
```
y[t]/.cvals[[1]]
1/4 t Sin[2t]
Plot[y[t]/.cvals, {t, 0, 16Pi}]
```

We verify the calculation with DSolve.

```
Clear[y]
DSolve[{y''[t] + 4y[t]==Cos[2t], y[0]==0, y'[0]==0},
y[t], t]/Simplify
{{y[t] -> 1/4 t Sin[2t]}}
```

Use Manipulate to help you see how changing parameter values and equations affect a system. With

```
Manipulate[
sol1 = DSolve[{mx''[t] + ax'[t] + kx[t]==Gamma Cos[omega t], x[0]==0, x'[0]==0}, x[t], t];
sol2 = NDSolve[{mx''[t] + ax'[t] + k Sin[x[t]]==Gamma Cos[omega t], x[0]==0, x'[0]==0},
x[t], {t, 0, 50}];
```

**FIGURE 6.16**

Comparing solutions of nonlinear initial-value problems to their corresponding linear approximations

```
p1 = Plot[x[t]/.sol1,{t,0,50},PlotRange→{-5,5},AspectRatio→1];
p2 = Plot[x[t]/.sol2,{t,0,50},PlotRange→{-5,5},AspectRatio→1];
Show[GraphicsRow[{p1,p2}],{{m,1},0,5},{{a,0},0,15,1},{{k,4},0,20,1},
{{ω,2},0,20,1},{{Γ,1},0,10,1}]
```

we can compare the solution of $mx'' + ax' + kx = \Gamma \cos \omega t$, $x(0) = 0$, $x'(0) = 0$ to the solution of $mx'' + ax' + k \sin x = \Gamma \cos \omega t$, $x(0) = 0$, $x'(0) = 0$ for various values of m , a , k , ω , and Γ . See Figure 6.16.

Example 6.2.6 (Hearing Beats and Resonance). In order to *hear* beats and resonance, we solve the initial-value problem

$$x'' + \omega^2 x = F \cos \beta t, \quad x(0) = \alpha, \quad x'(0) = \beta, \quad (6.13)$$

for each of the following parameter values: (a) $\omega^2 = 6000^2$, $\beta = 5991.62$, $F = 2$; and (b) $\omega^2 = 6000^2$, $\beta = 6000$, $F = 2$.

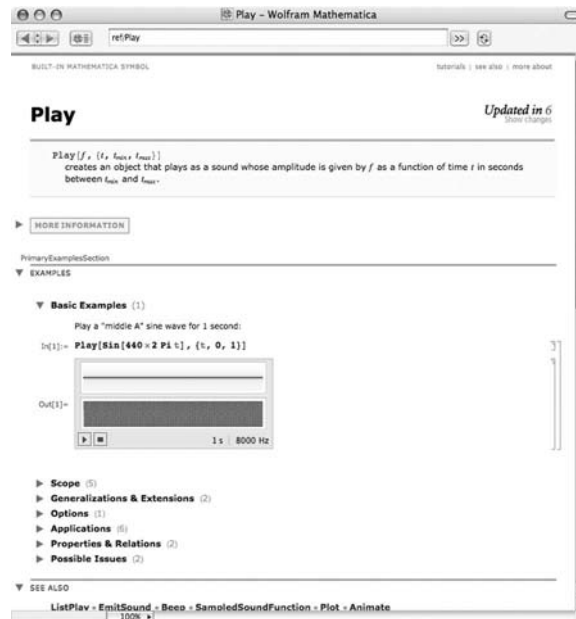
First, we define the function `sol` which, when given the parameters, solves the initial-value problem (6.13).

```
Clear[x, t, f, sol]
sol[ω_, β_, f_] := DSolve[{
  x''[t] + ω^2 x[t] == f Cos[β t], x[0] == 0, x'[0] == 0},
  x[t], t][[1, 1, 2]]
```

Thus, our solution for (a) is obtained by entering

```
a = sol[6000, 5991.62, 2]
-0.0000199025Cos[6000.t] + 0.0000198886Cos[8.38t]Cos[6000.t] +
  1.389859474088294*^-8Cos[6000.t]Cos[11991.6t] +
  0.0000198886Sin[8.38t]Sin[6000.t] +
  1.389859474088294*^-8Sin[6000.t]Sin[11991.6t]
```

To *hear* the function we use **Play** in the same way that we use **Plot** to see functions.



The values of **a** correspond to the amplitude of the sound as a function of time. See Figure 6.17(a).

Play[a, {t, 0, 6}]

Similarly, the solution for (b) is obtained by entering

```
b = sol[6000., 6000., 2]//Chop
-2.777*^-8Cos[6000.t]+2.777*^-8Cos[6000.t]^3+
  0.000166667 t Sin[6000.t]+1.3889*^-8 Sin[6000.t] Sin[12000.t]
```

We hear resonance with **Play**. See Figure 6.17(b).

Play[b, {t, 0, 6}]

**FIGURE 6.17**

Hearing and seeing beats and resonance: (a) Beats (b) Resonance

6.2.4 Variation of Parameters

A particular solution, y_p , is a solution that does not contain any arbitrary constants.

Observe that it is pointless to search for solutions of the form $y_p = c_1 y_1 + c_2 y_2$, where c_1 and c_2 are constants, because for every choice of c_1 and c_2 , $c_1 y_1 + c_2 y_2$ is a solution to the corresponding homogeneous equation.

Let $S = \{y_1, y_2\}$ be a fundamental set of solutions for equation (6.9). To solve the nonhomogeneous equation (6.8), we need to find a particular solution, y_p of equation (6.8). We search for a particular solution of the form

$$y_p = u_1(t)y_1(t) + u_2(t)y_2(t), \quad (6.14)$$

where u_1 and u_2 are functions of t . Differentiating equation (6.14) gives us

$$y_p' = u_1' y_1 + u_1 y_1' + u_2' y_2 + u_2 y_2'.$$

Assuming that

$$y_1 u_1' + y_2 u_2' = 0 \quad (6.15)$$

results in $y_p' = u_1 y_1' + u_2 y_2'$. Computing the second derivative then yields

$$y_p'' = u_1' y_1' + u_1 y_1'' + u_2' y_2' + u_2 y_2''.$$

Substituting y_p , y_p' , and y_p'' into equation (6.8) and using the facts that

$$u_1 (y_1'' + p y_1' + q y_1) = 0 \quad \text{and} \quad u_2 (y_2'' + p y_2' + q y_2) = 0$$

(because y_1 and y_2 are solutions to the corresponding homogeneous equation) results in

$$\begin{aligned} \frac{d^2 y_p}{dt^2} + p(t) \frac{dy_p}{dt} + q(t) y_p &= u_1' y_1' + u_1 y_1'' + u_2' y_2' + u_2 y_2'' + p(t) (u_1 y_1' + u_2 y_2') \\ &\quad + q(t) (u_1 y_1 + u_2 y_2) \\ &= y_1' u_1' + y_2' u_2' = f(t). \end{aligned} \quad (6.16)$$

Observe that equation (6.15) and equation (6.16) form a system of two linear equations in the unknowns u_1' and u_2' :

$$\begin{aligned} y_1 u_1' + y_2 u_2' &= 0 \\ y_1' u_1' + y_2' u_2' &= f(t). \end{aligned} \quad (6.17)$$

Applying Cramer's Rule gives us

$$u_1' = \frac{\begin{vmatrix} 0 & y_2' \\ f(t) & y_2' \end{vmatrix}}{\begin{vmatrix} y_1' & y_2' \\ y_1' & y_2' \end{vmatrix}} = -\frac{y_2(t)f(t)}{W(S)} \quad \text{and} \quad u_2' = \frac{\begin{vmatrix} y_1' & 0 \\ y_1' & f(t) \end{vmatrix}}{\begin{vmatrix} y_1' & y_2' \\ y_1' & y_2' \end{vmatrix}} = \frac{y_1(t)f(t)}{W(S)}, \quad (6.18)$$

where $W(S)$ is the Wronskian, $W(S) = \begin{vmatrix} y_1' & y_2' \\ y_1' & y_2' \end{vmatrix}$. After integrating to obtain u_1 and u_2 , we form y_p and then a general solution, $y = y_b + y_p$.

Example 6.2.7 Solve $y'' + 9y = \sec 3t$, $y(0) = 0$, $y'(0) = 0$, $0 \leq t < \pi/6$.

Solution The corresponding homogeneous equation is $y'' + 9y = 0$ with general solution $y_b = c_1 \cos 3t + c_2 \sin 3t$. Then, a fundamental set of solutions is $S = \{\cos 3t, \sin 3t\}$ and $W(S) = 3$, as we see using **Det**, and **Simplify**.

```
fs = {Cos[3t], Sin[3t]};
wm = {fs, D[fs, t]};
wm//MatrixForm
wd = Det[wm]//Simplify

$$\begin{pmatrix} \cos[3t] & \sin[3t] \\ -3\sin[3t] & 3\cos[3t] \end{pmatrix}$$

3
```

We use equation (6.18) to find $u_1 = \frac{1}{9} \ln \cos 3t$ and $u_2 = \frac{1}{3} t$.

```
u1 = Integrate[-Sin[3t]Sec[3t]/3, t]
u2 = Integrate[Cos[3t]Sec[3t]/3, t]

$$\frac{1}{9} \text{Log}[\cos[3t]]$$


$$\frac{1}{3} t$$

```

Absolute value is not needed in the antiderivatives because we are restricting the domain to $0 \leq t < \pi/6$ and $\cos t > 0$ on this interval.

It follows that a particular solution of the nonhomogeneous equation is $y_p = \frac{1}{9} \cos 3t \ln \cos 3t + \frac{1}{3} t \sin 3t$ and a general solution is $y = y_b + y_p = c_1 \cos 3t + c_2 \sin 3t + \frac{1}{9} \cos 3t \ln \cos 3t + \frac{1}{3} t \sin 3t$.

```
yp = u1Cos[3t] + u2Sin[3t]

$$\frac{1}{9} \cos[3t] \text{Log}[\cos[3t]] + \frac{1}{3} t \sin[3t]$$

```

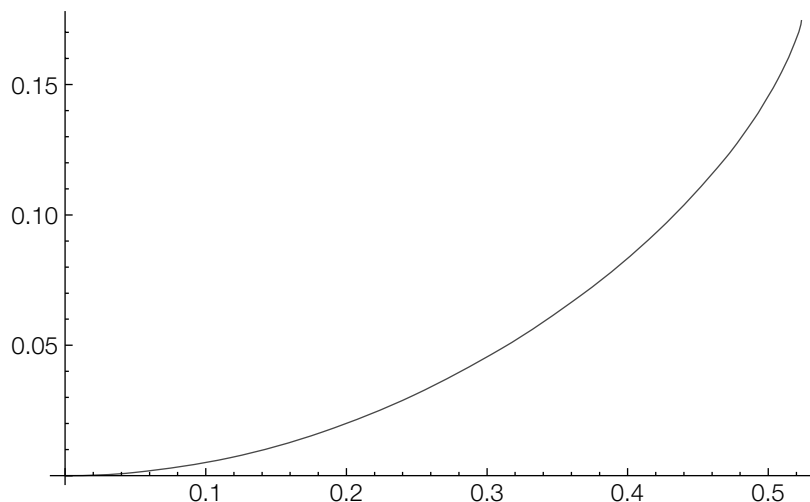
Identical results are obtained using **DSolve**.

```
DSolve[y''[t] + 9y[t] == Sec[3t], y[t], t]

$$\left\{ \left\{ y[t] \rightarrow C[1] \cos[3t] + C[2] \sin[3t] + \frac{1}{9} (\cos[3t] \text{Log}[\cos[3t]] + 3t \sin[3t]) \right\} \right\}$$

```

The negative sign in the output does not affect the result because $C[1]$ is arbitrary.

**FIGURE 6.18**

The domain of the solution is $0 \leq t < \pi/6$

Applying the initial conditions gives us $c_1 = c_2 = 0$, so we conclude that the solution to the initial-value problem is $y = \frac{1}{9} \cos 3t \ln \cos 3t + \frac{1}{3} t \sin 3t$.

```
sol = DSolve[{y''[t] + 9y[t]==Sec[3t], y[0]==0, y'[0]==0}, y[t], t]
{ {y[t] -> 1/9 (Cos[3t] Log[Cos[3t]] + 3t Sin[3t]) }
```

We graph the solution with **Plot** in Figure 6.18.

```
Plot[y[t]/.sol, {t, 0, Pi/6}]
```

6.3 HIGHER-ORDER LINEAR EQUATIONS

6.3.1 Basic Theory

The **standard form of the n th-order linear equation** is

$$\frac{d^n y}{dt^n} + a_{n-1}(t) \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1(t) \frac{dy}{dt} + a_0(t)y = f(t). \quad (6.19)$$

The **corresponding homogeneous equation** of equation (6.19) is

$$\frac{d^n y}{dt^n} + a_{n-1}(t) \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1(t) \frac{dy}{dt} + a_0(t)y = 0. \quad (6.20)$$

Let $y_1, y_2, \dots, y_n$ be n solutions of equation (6.20). The set $S = \{y_1, y_2, \dots, y_n\}$ is **linearly independent** if and only if the **Wronskian**,

$$W(S) = \begin{vmatrix} y_1 & y_2 & y_3 & \cdots & y_n \\ y_1' & y_2' & y_3' & \cdots & y_n' \\ y_1'' & y_2'' & y_3'' & \cdots & y_n'' \\ y_1^{(3)} & y_2^{(3)} & y_3^{(3)} & \cdots & y_n^{(3)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_1^{(n-1)} & y_2^{(n-1)} & y_3^{(n-1)} & \cdots & y_n^{(n-1)} \end{vmatrix} \quad (6.21)$$

is not identically the zero function. S is **linearly dependent** if S is not linearly independent.

If $y_1, y_2, \dots, y_n$ are n linearly independent solutions of equation (6.20), we say that $S = \{y_1, y_2, \dots, y_n\}$ is a **fundamental set** for equation (6.20), and a **general solution** of equation (6.20) is $y = c_1 y_1 + c_2 y_2 + c_3 y_3 + \cdots + c_n y_n$.

A **general solution** of equation (6.19) is $y = y_b + y_p$, where y_b is a general solution of the corresponding homogeneous equation and y_p is a particular solution of equation (6.19).

6.3.2 Constant Coefficients

If

$$\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = 0$$

has real constant coefficients, we assume that $y = e^{kt}$ and find that k satisfies the **characteristic equation**

$$k^n + a_{n-1} k^{n-1} + \cdots + a_1 k + a_0 = 0. \quad (6.22)$$

If a solution k of equation (6.22) has multiplicity m , m linearly independent solutions corresponding to k are

$$e^{kt}, te^{kt}, \dots, t^{m-1} e^{kt}.$$

If a solution $k = \alpha + \beta i$, $\beta \neq 0$, of equation (6.22) has multiplicity m , $2m$ linearly independent solutions corresponding to $k = \alpha + \beta i$ (and $k = \alpha - \beta i$) are

$$e^{\alpha t} \cos \beta t, e^{\alpha t} \sin \beta t, te^{\alpha t} \cos \beta t, te^{\alpha t} \sin \beta t, \dots, t^{m-1} e^{\alpha t} \cos \beta t, t^{m-1} e^{\alpha t} \sin \beta t.$$

Example 6.3.1 Solve $12y''' - 5y'' - 6y' - y = 0$.

Solution The characteristic equation is

$$12k^3 - 5k^2 - 6k - 1 = (k - 1)(3k + 1)(4k + 1) = 0$$

Factor[expression]
attempts to factor
expression.

with solutions $k_1 = -1/3$, $k_2 = -1/4$, and $k_3 = 1$.

```
Factor[12k^3-5k^2-6k-1]
(-1+k)(1+3k)(1+4k)
```

Thus, three linearly independent solutions of the equation are $y_1 = e^{-t/3}$, $y_2 = e^{-t/4}$, and $y_3 = e^t$; a general solution is $y = c_1 e^{-t/3} + c_2 e^{-t/4} + c_3 e^t$. We check with DSolve.

```
Clear[y]
DSolve[12y'''[t]-5y''[t]-6y'[t]-y[t]==0, y[t], t]
{{y[t] -> e^{-t/4}C[1] + e^{-t/3}C[2] + e^tC[3]}}
```

Example 6.3.2 Solve $y''' + 4y' = 0$, $y(0) = 0$, $y'(0) = 1$, $y''(0) = -1$.

Solution The characteristic equation is $k^3 + 4k = k(k^2 + 4) = 0$ with solutions $k_1 = 0$ and $k_{2,3} = \pm 2i$ that are found with Solve.

Enter ?Solve to obtain basic help regarding the Solve function.

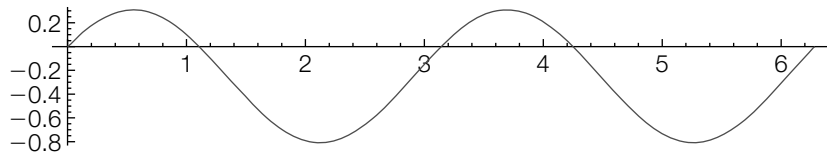
```
Solve[k^3+4k==0]
{{k -> 0}, {k -> -2i}, {k -> 2i}}
```

Three linearly independent solutions of the equation are $y_1 = 1$, $y_2 = \cos 2t$, and $y_3 = \sin 2t$. A general solution is $y = c_1 + c_2 \sin 2t + c_3 \cos 2t$.

```
gensol = DSolve[y'''[t] + 4y'[t]==0, y[t], t]
{{y[t] -> C[3] - 1/2 C[2] Cos[2t] + 1/2 C[1] Sin[2t]}}
```

Application of the initial conditions shows us that $c_1 = -1/4$, $c_2 = 1/2$, and $c_3 = 1/4$, so the solution to the initial-value problem is $y = -\frac{1}{4} + \frac{1}{2} \sin 2t + \frac{1}{4} \cos 2t$. We verify the computation with DSolve and graph the result with Plot in Figure 6.19.

```
e1 = y[t]/.gensol[[1]]/.t -> 0
- C[2]/2 + C[3]
e2 = D[y[t]/.gensol[[1]], t]/.t -> 0
e3 = D[y[t]/.gensol[[1]], {t, 2}]/.t -> 0
C[1]
2C[2]
cvals = Solve[{e1==0, e2==1, e3== -1}]
{{C[1] -> 1, C[2] -> -1/2, C[3] -> -1/4}}
Clear[y]
partsol = DSolve[{y'''[t] + 4y'[t]==0, y[0]==0,
y'[0]==1, y''[0]== -1}, y[t], t]
{{y[t] -> 1/4 (-1 + Cos[2t] + 2Sin[2t])}}
Plot[y[t]/.partsol, {t, 0, 2Pi}, AspectRatio -> Automatic]
```

**FIGURE 6.19**Graph of $y = -\frac{1}{4} + \frac{1}{2} \sin 2t + \frac{1}{4} \cos 2t$

Example 6.3.3 Find a differential equation with general solution $y = c_1 e^{-2t/3} + c_2 t e^{-2t/3} + c_3 t^2 e^{-2t/3} + c_4 \cos t + c_5 \sin t + c_6 t \cos t + c_7 t \sin t + c_8 t^2 \cos t + c_9 t^2 \sin t$.

Solution A linear homogeneous differential equation with constant coefficients that has this general solution has fundamental set of solutions

$$S = \{e^{-2t/3}, t e^{-2t/3}, t^2 e^{-2t/3}, \cos t, \sin t, t \cos t, t \sin t, t^2 \cos t, t^2 \sin t\}.$$

Hence, in the characteristic equation $k = -2/3$ has multiplicity 3, whereas $k = \pm i$ has multiplicity 3. The characteristic equation is

$$27 \left(k + \frac{2}{3}\right)^3 (k - i)^3 (k + i)^3 = k^9 + 2k^8 + \frac{13}{3}k^7 + \frac{170}{27}k^6 + 7k^5 + \frac{62}{9}k^4 + 5k^3 + \frac{26}{9}k^2 + \frac{4}{3}k + \frac{8}{27},$$

where we use Mathematica to compute the multiplication with **Expand**.

Expand[27(k + 2/3)^3(k^2 + 1)^3]

$$8 + 36k + 78k^2 + 135k^3 + 186k^4 + 189k^5 + 170k^6 + 117k^7 + 54k^8 + 27k^9$$

Thus, a differential equation obtained after dividing by 27 with the indicated general solution is

$$\begin{aligned} \frac{d^9 y}{dt^9} + 2 \frac{d^8 y}{dt^8} + \frac{13}{3} \frac{d^7 y}{dt^7} + \frac{170}{27} \frac{d^6 y}{dt^6} + 7 \frac{d^5 y}{dt^5} + \frac{62}{9} \frac{d^4 y}{dt^4} \\ + 5 \frac{d^3 y}{dt^3} + \frac{26}{9} \frac{d^2 y}{dt^2} + \frac{4}{3} \frac{dy}{dt} + \frac{8}{27} y = 0. \end{aligned}$$

6.3.3 Undetermined Coefficients

For higher-order linear equations with constant coefficients, the method of undetermined coefficients is the same as for second-order equations discussed in Section 6.2.3, provided that the forcing function involves the terms discussed in Section 6.2.3.

Example 6.3.4 Solve $\frac{d^3 y}{dt^3} + \frac{2}{3} \frac{d^2 y}{dt^2} + \frac{145}{9} \frac{dy}{dt} = e^{-t}$, $y(0) = 1$, $\frac{dy}{dt}(0) = 2$, $\frac{d^2 y}{dt^2}(0) = -1$.

Solution The corresponding homogeneous equation, $y''' + \frac{2}{3}y'' + \frac{145}{9}y' = 0$, has general solution $y_h = c_1 + (c_2 \sin 4t + c_3 \cos 4t) e^{-t/3}$ and a fundamental set of solutions for the corresponding homogeneous equation is $S = \{1, e^{-t/3} \cos 4t, e^{-t/3} \sin 4t\}$.

```
DSolve[y'''[t] + 2/3 y''[t] + 145/9 y'[t] == 0,
y[t], t]
{ {y[t] -> C[3] - 3/145 e^{-t/3}
((12 C[1] + C[2]) Cos[4 t] + (C[1] - 12 C[2]) Sin[4 t])} }
```

For e^{-t} , the associated set of functions is $F = \{e^{-t}\}$. Because no element of F is an element of S , we assume that $y_p = A e^{-t}$, where A is a constant to be determined. After defining y_p , we compute the necessary derivatives

```
Clear[yp]
yp[t_] = a Exp[-t];
yp'[t]
yp''[t]
yp'''[t]
- a e^{-t}
a e^{-t}
- a e^{-t}
```

and substitute into the nonhomogeneous equation.

```
eqn = yp'''[t] + 2/3 yp''[t] + 145/9 yp'[t] == Exp[-t]
- 148/9 a e^{-t} == e^{-t}
```

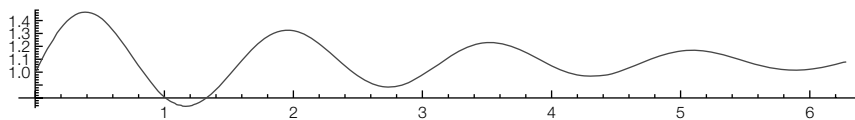
Equating coefficients and solving for A gives us $A = -9/148$ so $y_p = -\frac{9}{148} e^{-t}$ and a general solution is $y = y_h + y_p$.

Remark 6.3 `SolveAlways[equation, variable]` attempts to solve an equation so that it is true for all values of variable.

```
SolveAlways[eqn, t]
{ {a -> -9/148} }
```

We verify the result with `DSolve`.

```
gensol = DSolve[y'''[t] + 2/3 y''[t] +
145/9 y'[t] == Exp[-t], y[t], t] // FullSimplify
{ {y[t] -> -9 e^{-t}/148 + C[3] - 3/145 e^{-t/3} ((12 C[1] + C[2]) Cos[4 t] +
(C[1] - 12 C[2]) Sin[4 t])} }
```

**FIGURE 6.20**

The solution of the equation that satisfies $y(0) = 1$, $y'(0) = 2$, and $y''(0) = -1$

To obtain a real-valued solution, we use **ComplexExpand**. If you are using a version of Mathematica older than version 6, you might receive a complex valued function rather than the real-valued function that we obtained. In those cases, **ComplexExpand** can help you rewrite your complex solution as a real-valued solution.

To apply the initial conditions, we compute $y(0) = 1$, $y'(0) = 2$, and $y''(0) = -1$ and solve for c_1 , c_2 , and c_3 . The solution of the initial-value problem is obtained by substituting these values into the general solution, and then we graph the result with **Plot** in Figure 6.20.

```

initsol = DSolve[{y'''[t] + 2/3 y''[t] +
145/9 y'[t] == Exp[-t], y[0] == 1, y'[0] == 2, y''[0] == -1},
y[t], t] // FullSimplify
{ { { y[t] -> (e^{-t} (-2610 + 46472 e^1 + e^{2t/3} (-942 Cos[4t] + 20729 Sin[4t])) ) / 42920 } } }
Plot[y[t]/.initsol, {t, 0, 2Pi}, AspectRatio -> Automatic]

```

Example 6.3.5 Solve

$$\frac{d^8 y}{dt^8} + \frac{7}{2} \frac{d^7 y}{dt^7} + \frac{73}{2} \frac{d^6 y}{dt^6} + \frac{229}{2} \frac{d^5 y}{dt^5} + \frac{801}{2} \frac{d^4 y}{dt^4} + 976 \frac{d^3 y}{dt^3} + 1168 \frac{d^2 y}{dt^2} + 640 \frac{dy}{dt} + 128y = te^{-t} + \sin 4t + t.$$

Solution Solving the characteristic equation

```

Solve[k^8 + 7/2 k^7 + 73/2 k^6 + 229/2 k^5 +
801/2 k^4 + 976 k^3 + 1168 k^2 + 640 k + 128 == 0]
{ {k -> -1/2}, {k -> -1}, {k -> -1}, {k -> -1/2},
{k -> -4i}, {k -> -4i}, {k -> 4i}, {k -> 4i} }

```

shows us that the solutions are $k_1 = -1/2$, $k_2 = -1$ with multiplicity 3, and $k_{3,4} = \pm 4i$, each with multiplicity 2. A fundamental set of solutions for the corresponding homogeneous equation is

$$S = \{e^{-t/2}, e^{-t}, te^{-t}, t^2 e^{-t}, \cos 4t, t \cos 4t, \sin 4t, t \sin 4t\}.$$

A general solution of the corresponding homogeneous equation is

$$y_h = c_1 e^{-t/2} + (c_2 + c_3 t + c_4 t^2) e^{-t} + (c_5 + c_7 t) \sin 4t + (c_6 + c_8 t) \cos 4t.$$

```
gensol = DSolve[D[y[t], {t, 8}] + 7/2D[y[t], {t, 7}] +
73/2D[y[t], {t, 6}] + 229/2D[y[t], {t, 5}] +
801/2D[y[t], {t, 4}] + 976y'''[t] + 1168y''[t] +
640y'[t] + 128y[t]==0, y[t], t]
```

```
{ {y[t] -> e^{-t/2} C[5] + e^{-t} C[6] + e^{-t} t C[7] + e^{-t} t^2 C[8] +
C[1] Cos[4t] + t C[2] Cos[4t] + C[3] Sin[4t] + t C[4] Sin[4t]} }
```

The associated set of functions for te^{-t} is $F_1 = \{e^{-t}, te^{-t}\}$. We multiply F_1 by t^n , where n is the smallest nonnegative integer so that no element of $t^n F_1$ is an element of S : $t^3 F_1 = \{t^3 e^{-t}, t^4 e^{-t}\}$. The associated set of functions for $\sin 4t$ is $F_2 = \{\cos 4t, \sin 4t\}$. We multiply F_2 by t^n , where n is the smallest nonnegative integer so that no element of $t^n F_2$ is an element of S : $t^2 F_2 = \{t^2 \cos 4t, t^2 \sin 4t\}$. The associated set of functions for t is $F_3 = \{1, t\}$. No element of F_3 is an element of S .

Thus, we search for a particular solution of the form

$$y_p = A_1 t^3 e^{-t} + A_2 t^4 e^{-t} + A_3 t^2 \cos 4t + A_4 t^2 \sin 4t + A_5 + A_6 t,$$

where the A_i are constants to be determined.

After defining y_p ,

```
yp[t_] = a[1]t^3Exp[-t] + a[2]t^4Exp[-t] +
a[3]t^2Cos[4t] + a[4]t^2Sin[4t] + a[5] + a[6]t;
```

we substitute into the nonhomogeneous equation, naming the result `eqn`. At this point we can either equate coefficients and solve for A_i or use the fact that `eqn` is true for *all* values of t .

```
eqn = D[yp[t], {t, 8}] + 7/2D[yp[t], {t, 7}] +
73/2D[yp[t], {t, 6}] + 229/2D[yp[t], {t, 5}] +
801/2D[yp[t], {t, 4}] + 976yp'''[t] + 1168yp''[t] +
640yp'[t] + 128yp[t]==tExp[-t] + Sin[4t] + t//Simplify
```

```
e^{-t} (-867a[1] + 7752a[2] - 3468ta[2] + 128e^ta[5] +
640e^ta[6] + 128e^ta[6] - 64e^t(369a[3] - 428a[4])Cos[4t] -
64e^t(428a[3] + 369a[4])Sin[4t]) == t + e^{-t} t + Sin[4t]
```

We substitute in six values of t

```
sysofeqs = Table[eqn/.t -> n//N, {n, 0, 5}];
```

and then solve for A_i .

```
coeffs = Solve[sysofeqs, {a[1.], a[2.], a[3.], a[4.], a[5.], a[6.]}]
{{a[1.] -> -0.00257819, a[2.] -> -0.000288351, a[3.] -> -0.0000209413,
a[4.] -> -0.0000180545, a[5.] -> -0.0390625, a[6.] -> 0.0078125}}
```

y_p is obtained by substituting the values for A_i into y_p and a general solution is $y = y_h + y_p$. DSolve is able to find an exact solution.

```
gensol = DSolve[D[y[t], {t, 8}] + 7/2D[y[t], {t, 7}] +
73/2D[y[t], {t, 6}] + 229/2D[y[t], {t, 5}] +
801/2D[y[t], {t, 4}] + 976y'''[t] + 1168y''[t] +
640y'[t] + 128y[t]==tExp[-t] + Sin[4t] + t, y[t], t]/Simplify
```

```
{ {y[t] -> 1/40727223623424000 e^{-t}
(4394000 (72412707 e^{t(-5+t)} + 9268826496 e^{1/2 C[5]} - 32(35097672 +
746776 t^3 + 83521 t^4 - 289650828 C[6] - 204 t(-86016 + 1419857 C[7]) -
3468 t^2(-1270 + 83521 C[8])) -
204 e^{t(-9041976373 + 4180789600 t^2 - 199643253056000 C[1] -
4420 t(-1568449 + 45168156800 C[2])) Cos[4t] -
51 e^{t(-13794625331 + 14417863200 t^2 - 798573012224000 C[3] -
2263040 t(20406 + 352876225 C[4])) Sin[4t])} }
```

Variation of Parameters

In the same way as with second-order equations, we assume that a particular solution of the n th-order linear equation (6.19) has the form $y_p = u_1(t)y_1 + u_2(t)y_2 + \cdots + u_n(t)y_n$, where $S = \{y_1, y_2, \dots, y_n\}$ is a fundamental set of solutions to the corresponding homogeneous equation (6.20). With the assumptions

$$\begin{aligned} y_p' &= y_1 u_1' + y_2 u_2' + \cdots + y_n u_n' = 0 \\ y_p'' &= y_1' u_1' + y_2' u_2' + \cdots + y_n' u_n' = 0 \\ &\vdots \\ y_p^{(n-1)} &= y_1^{(n-2)} u_1' + y_2^{(n-2)} u_2' + \cdots + y_n^{(n-2)} u_n' = 0, \end{aligned} \quad (6.23)$$

we obtain the equation

$$y_1^{(n-1)} u_1' + y_2^{(n-1)} u_2' + \cdots + y_n^{(n-1)} u_n' = f(t). \quad (6.24)$$

Equations (6.23) and (6.24) form a system of n linear equations in the unknowns $u_1', u_2', \dots, u_n'$. Applying Cramer's rule,

$$u_i' = \frac{W_i(S)}{W(S)}, \quad (6.25)$$

where $W(S)$ is given by equation (6.21) and $W_i(S)$ is the determinant of the matrix obtained by replacing the i th column of

$$\begin{pmatrix} y_1 & y_2 & \cdots & y_n \\ y_1' & y_2' & \cdots & y_n' \\ \vdots & \vdots & \cdots & \vdots \\ y_1^{(n-1)} & y_2^{(n-1)} & \cdots & y_n^{(n-1)} \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 0 \\ 0 \\ \vdots \\ f(t) \end{pmatrix}.$$

Example 6.3.6 Solve $y^{(3)} + 4y' = \sec 2t$.

Solution A general solution of the corresponding homogeneous equation is $y_h = c_1 + c_2 \cos 2t + c_3 \sin 2t$; a fundamental set is $S = \{1, \cos 2t, \sin 2t\}$ with Wronskian $W(S) = 8$.

```

yh = DSolve[y'''[t] + 4y'[t]==0, y[t], t]
{ {y[t] -> C[3] - 1/2 C[2] Cos[2t] + 1/2 C[1] Sin[2t]} }

s = {1, Cos[2t], Sin[2t]};
ws = {s, D[s, t], D[s, {t, 2}]};
MatrixForm[ws]

$$\begin{pmatrix} 1 & \cos[2t] & \sin[2t] \\ 0 & -2\sin[2t] & 2\cos[2t] \\ 0 & -4\cos[2t] & -4\sin[2t] \end{pmatrix}$$

dws = Simplify[Det[ws]]
8

```

Using variation of parameters to find a particular solution of the nonhomogeneous equation, we let $y_1 = 1$, $y_2 = \cos 2t$, and $y_3 = \sin 2t$ and assume that a particular solution has the form $y_p = u_1 y_1 + u_2 y_2 + u_3 y_3$. Using the variation of parameters formula, we obtain

$$u'_1 = \frac{1}{8} \begin{vmatrix} 0 & \cos 2t & \sin 2t \\ 0 & -2\sin 2t & 2\cos 2t \\ \sec 2t & -4\cos 2t & -4\sin 2t \end{vmatrix} = \frac{1}{4} \sec 2t \quad \text{so} \quad u_1 = \frac{1}{8} \ln |\sec 2t + \tan 2t|,$$

$$u'_2 = \frac{1}{8} \begin{vmatrix} 1 & 0 & \sin 2t \\ 0 & 0 & 2\cos 2t \\ 0 & \sec 2t & -4\sin 2t \end{vmatrix} = -\frac{1}{4} \quad \text{so} \quad u_2 = -\frac{1}{4}t,$$

and

$$u'_3 = \frac{1}{8} \begin{vmatrix} 1 & \cos 2t & 0 \\ 0 & -2\sin 2t & 0 \\ 0 & -4\cos 2t & \sec 2t \end{vmatrix} = -\frac{1}{2} \tan 2t \quad \text{so} \quad u_3 = \frac{1}{8} \ln |\cos 2t|,$$

where we use **Det** and **Integrate** to evaluate the determinants and integrals. In the case of u_1 , the output given by Mathematica looks different than the result we obtained by hand, but differentiating the difference between the two results yields 0, so the results obtained by hand and with Mathematica are the same.

```

u1p = 1/8 Det[{{0, Cos[2t], Sin[2t]}, {0, -2Sin[2t], 2Cos[2t]},
{Sec[2t], -4Cos[2t], -4Sin[2t]}}]//Simplify

```

```

1/4 Sec[2t]

```

```

u1 = Integrate[u1p, t]

```

```

1/4 ArcTanh[Tan[t]]

```

```

s1 = D[u1 - 1/8 Log[Sec[2t] + Tan[2t]], t]

```

```

Sec[t]^2 / (4 (1 - Tan[t]^2)) - (2 Sec[2t]^2 + 2 Sec[2t] Tan[2t]) / (8 (Sec[2t] + Tan[2t]))

```

```

Simplify[s1]
0

u2p = 1/8Det[{{1, 0, Sin[2t]}, {0, 0, 2Cos[2t]},
             {0, Sec[2t], -4Sin[2t]}}]/Simplify

-1/4

u2 = Integrate[u2p, t]
-1/4

u3p = 1/8Det[{{1, Cos[2t], 0}, {0, -2Sin[2t], 0},
             {0, -4Cos[2t], Sec[2t]}}]/Simplify

-1/4 Tan[2t]

u3 = Integrate[u3p, t]
1/8 Log[Cos[2t]]

```

Thus, a particular solution of the nonhomogeneous equation is

$$y_p = \frac{1}{8} \ln |\sec 2t + \tan 2t| - \frac{1}{4} t \cos 2t + \frac{1}{8} \ln |\cos 2t| \sin 2t$$

and a general solution is $y = y_b + y_p$. We verify that the calculations using DSolve return an equivalent solution.

```

gensol = DSolve[y'''[t] + 4y'[t] == Sec[2t], y[t], t]/Simplify
{ {y[t] -> 1/8 (2ArcTanh[Tan[t]] + 8C[3] -
               8C[2]Cos[t]^2 - 2tCos[2t] + 4C[1]Sin[2t] + Log[Cos[2t]]Sin[2t]) } }

```

6.3.4 Laplace Transform Methods

The *method of Laplace transforms* can be useful when the forcing function is piecewise-defined or periodic.

Definition 3 (Laplace Transform and Inverse Laplace Transform). Let $y = f(t)$ be a function defined on the interval $[0, \infty)$. The **Laplace transform** is the function (of s)

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt, \quad (6.26)$$

provided the improper integral exists. $f(t)$ is the **inverse Laplace transform** of $F(s)$ means that $\mathcal{L}\{f(t)\} = F(s)$ and we write $\mathcal{L}^{-1}\{F(s)\} = f(t)$.

1. LaplaceTransform[f[t],t,s] computes $\mathcal{L}\{f(t)\} = F(s)$.
2. InverseLaplaceTransform[F[s],t,s] computes $\mathcal{L}^{-1}\{F(s)\} = f(t)$.
3. UnitStep[t] returns $\mathcal{U}(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0. \end{cases}$

Typically, when we use Laplace transforms to solve a differential equation for a function $y(t)$, we will compute the Laplace transform of each term of the differential equation, solve the resulting algebraic equation for the Laplace transform of $y(t)$, $\mathcal{L}\{y(t)\} = Y(s)$, and, finally, determine $y(t)$ by computing the inverse Laplace transform of $Y(s)$, $\mathcal{L}^{-1}\{Y(s)\} = y(t)$.

Example 6.3.7 Let $y = f(t)$ be defined recursively by $f(t) = \begin{cases} 1, & 0 \leq t < 1 \\ -1, & 1 \leq t < 2 \end{cases}$ and $f(t) = f(t - 2)$ if $t \geq 2$. Solve $y'' + 4y' + 20y = f(t)$.

Solution We begin by defining and graphing $y = f(t)$ for $0 \leq t \leq 5$ in Figure 6.21.

```
Clear[f, g, u, y1, y2, sol]
f[t_]:=1/;0 ≤ t < 1
f[t_]:= -1/;1 ≤ t ≤ 2
f[t_]:=f[t-2]/;t > 2
Plot[f[t], {t, 0, 5}, Ticks → {Automatic, {-2, -1, 0, 1, 2}},
PlotRange → {-2, 2}]
```

We then define lhs to be the left-hand side of the equation $y'' + 4y' + 20y = f(t)$,

```
Clear[y, x, lhs, steptone, steptwo]
lhs = y''[t] + 4y'[t] + 20y[t];
```

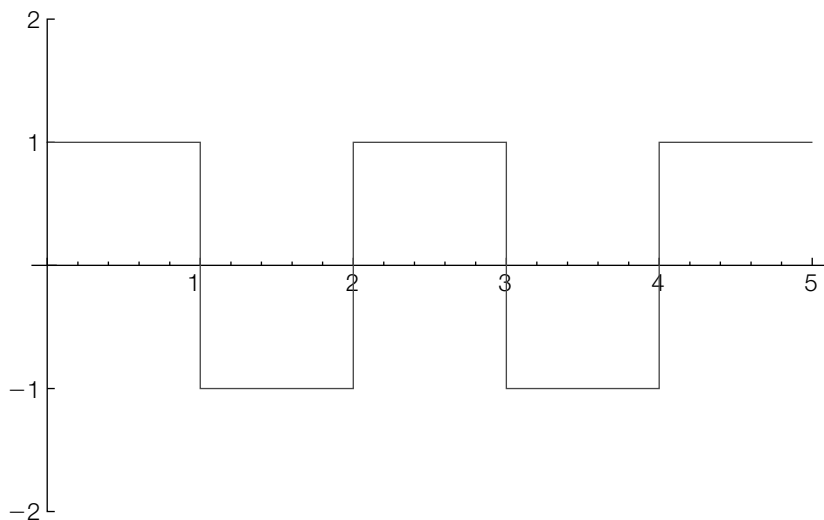


FIGURE 6.21

Plot of $f(t)$ for $0 \leq t \leq 5$

and compute the Laplace transform of lhs with `LaplaceTransform`, naming the result `stepone`.

stepone = LaplaceTransform[lhs, t, s]

`20LaplaceTransform[y[t], t, s] + s2LaplaceTransform[y[t], t, s] +
(4(s LaplaceTransform[y[t], t, s] - y[0]) - sy[0] - y'[0])`

Let `lr` denote the Laplace transform of the right-hand side of the equation, $f(t)$. We now solve the equation $20ly + 4sly + s^2ly - 4y(0) - sy(0) - y'(0) = lr$ for `ly` and name the resulting output `steptwo`.

steptwo = Solve[stepone == lr, LaplaceTransform[y[t], t, s]]

`{ { LaplaceTransform[y[t], t, s] ->  $\frac{lr + 4y[0] + sy[0] + y'[0]}{20 + 4s + s^2}$  } }`

stepthree = ExpandNumerator[steptwo[[1, 1, 2]], 1r]

`$\frac{lr + 4y[0] + sy[0] + y'[0]}{20 + 4s + s^2}$`

To find $y(t)$, we must compute the inverse Laplace transform of $\mathcal{L}\{y(t)\}$; the formula for which is explicitly obtained from `steptwo` with `steptwo[[1, 1, 2]]`. First, we rewrite : $\mathcal{L}\{y(t)\}$. Then,

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ \frac{\mathcal{L}\{f(t)\}}{s^2 + 4s + 20} + \frac{4y(0) + sy(0) + y'(0)}{s^2 + 4s + 20} \right\} \\ &= \mathcal{L}^{-1} \left\{ \frac{\mathcal{L}\{f(t)\}}{s^2 + 4s + 20} \right\} + \mathcal{L}^{-1} \left\{ \frac{4y(0) + sy(0) + y'(0)}{s^2 + 4s + 20} \right\}. \end{aligned}$$

Completing the square yields $s^2 + 4s + 20 = (s + 2)^2 + 16$. Because

$$\mathcal{L}^{-1} \left\{ \frac{b}{(s - a)^2 + b^2} \right\} = e^{at} \sin bt \quad \text{and} \quad \mathcal{L}^{-1} \left\{ \frac{s - a}{(s - a)^2 + b^2} \right\} = e^{at} \cos bt,$$

the inverse Laplace transform of

$$\frac{4y(0) + sy(0) + y'(0)}{s^2 + 4s + 20} = y(0) \frac{s + 2}{(s + 2)^2 + 4^2} + \frac{y'(0) + 2y(0)}{4} \frac{4}{(s + 2)^2 + 4^2}$$

is

$$y(0)e^{-2t} \cos 4t + \frac{y'(0) + 2y(0)}{4} e^{-2t} \sin 4t,$$

which is defined as $y_1(t)$. We perform these steps with Mathematica by first using `InverseLaplaceTransform` to calculate $\mathcal{L}^{-1} \left\{ \frac{4y(0) + sy(0) + y'(0)}{s^2 + 4s + 20} \right\}$, naming the result `stepfour`.

stepfour = InverseLaplaceTransform `$\left[ -\frac{-4y[0] - sy[0] - y'[0]}{20 + 4s + s^2}, s, t \right]$`
 `$-\frac{1}{8}e^{(-2-4i)t}((-2+4i)y[0] + (2+4i)e^{8it}y[0] - y'[0] + e^{8it}y'[0])$`

To see that this is a real-valued function, we use **ComplexExpand** together with **Simplify**.

stepfive = ComplexExpand[stepfour]//Simplify

$$\frac{1}{4}e^{-2t} (4\cos[4t]y[0] + \sin[4t] (2y[0] + y'[0]))$$

If the result in **stepfive** is given in terms of real and imaginary parts of $y(0)$ and $y'(0)$, because $y'(0)$ is assumed to be a real number, the imaginary part of $y'(0)$ is 0; the real part of $y'(0)$ is $y'(0)$.

y1[t_] = stepfive/.{Im[y'[0]] → 0, Re[y'[0]] → y'[0]}//Simplify

$$\frac{1}{4}e^{-2t} (4\cos[4t]y[0] + \sin[4t] (2y[0] + y'[0]))$$

To compute the inverse Laplace transform of $\frac{\mathcal{L}\{f(t)\}}{s^2 + 4s + 20}$, we begin by computing

$\text{lr} = \mathcal{L}\{f(t)\}$. Let $\mathcal{U}_a(t) = \begin{cases} 1, & t \geq a \\ 0, & t < a \end{cases}$. Then, $\mathcal{U}_a(t) = \mathcal{U}(t - a) = \text{UnitStep}[t - a]$.

The periodic function $f(t) = \begin{cases} 1, & 0 \leq t < 1 \\ -1, & 1 \leq t < 2 \end{cases}$ and $f(t) = f(t - 2)$ if $t \geq 2$ can be written in terms of step functions as

$$\begin{aligned} f(t) &= \mathcal{U}_0(t) - 2\mathcal{U}_1(t) + 2\mathcal{U}_2(t) - 2\mathcal{U}_3(t) + 2\mathcal{U}_4(t) - \dots \\ &= \mathcal{U}(t) - 2\mathcal{U}(t - 1) + 2\mathcal{U}(t - 2) - 2\mathcal{U}(t - 3) + 2\mathcal{U}(t - 4) - \dots \\ &= \mathcal{U}(t) + 2 \sum_{n=1}^{\infty} (-1)^n \mathcal{U}(t - n). \end{aligned}$$

The Laplace transform of $\mathcal{U}_a(t) = \mathcal{U}(t - a)$ is $\frac{1}{s}e^{-as}$ and the Laplace transform of $f(t)\mathcal{U}_a(t) = f(t)\mathcal{U}(t - a)$ is $e^{-as}F(s)$, where $F(s)$ is the Laplace transform of $f(t)$. Then,

$$\begin{aligned} \text{lr} &= \frac{1}{s} - \frac{2}{s}e^{-s} + \frac{2}{s}e^{-2s} - \frac{2}{s}e^{-3s} + \dots \\ &= \frac{1}{s} (1 - 2e^{-s} + 2e^{-2s} - 2e^{-3s} + \dots) \end{aligned}$$

and

$$\begin{aligned} \frac{\text{lr}}{s^2 + 4s + 20} &= \frac{1}{s(s^2 + 4s + 20)} (1 - 2e^{-s} + 2e^{-2s} - 2e^{-3s} + \dots) \\ &= \frac{1}{s(s^2 + 4s + 20)} + 2 \sum_{n=1}^{\infty} (-1)^n \frac{e^{-ns}}{s(s^2 + 4s + 20)}. \end{aligned}$$

Because $\frac{1}{s^2 + 4s + 20} = \frac{1}{4} \frac{1}{(s + 2)^2 + 4^2}$, $\mathcal{L}^{-1} \left\{ \frac{1}{s(s^2 + 4s + 20)} \right\} = \int_0^t \frac{1}{4} e^{-2\alpha}$

$\sin 4\alpha \, d\alpha$, computed and defined to be the function $g(t)$.

g[t_] = Integrate[Exp[-2α]Sin[4α] dα, {α, 0, t}]

$$\frac{1}{40} (2 - e^{-2t} (2\cos[4t] + \sin[4t]))$$

Alternatively, we can use `InverseLaplaceTransform` to obtain the same result.

```
g[t_]=InverseLaplaceTransform  $\left[\frac{1}{s(s^2+4s+20)}, s, t\right]$  //ExpToTrig//  
Simplify
```

$$\frac{1}{80}(4 + (2\cos[4t] + \sin[4t])(-2\cosh[2t] + 2\sinh[2t]))$$

Then, $\mathcal{L}^{-1}\left\{2(-1)^n \frac{e^{-ns}}{s(s^2+4s+20)}\right\} = 2(-1)^n g(t-n)\mathcal{U}(t-n)$ and the inverse Laplace transform of

$$\frac{1}{s(s^2+4s+20)} + 2 \sum_{n=1}^{\infty} (-1)^n \frac{e^{-ns}}{s(s^2+4s+20)}$$

is

$$y_2(t) = g(t) + 2 \sum_{n=1}^{\infty} (-1)^n g(t-n)\mathcal{U}(t-n).$$

It then follows that

$$y'(t) = y_1'(t) + y_2'(t) \\ = y'(0)e^{-2t} \cos 4t + \frac{y'(0) + 2y(0)}{4} e^{-2t} \sin 4t + 2 \sum_{n=1}^{\infty} (-1)^n g(t-n)\mathcal{U}(t-n),$$

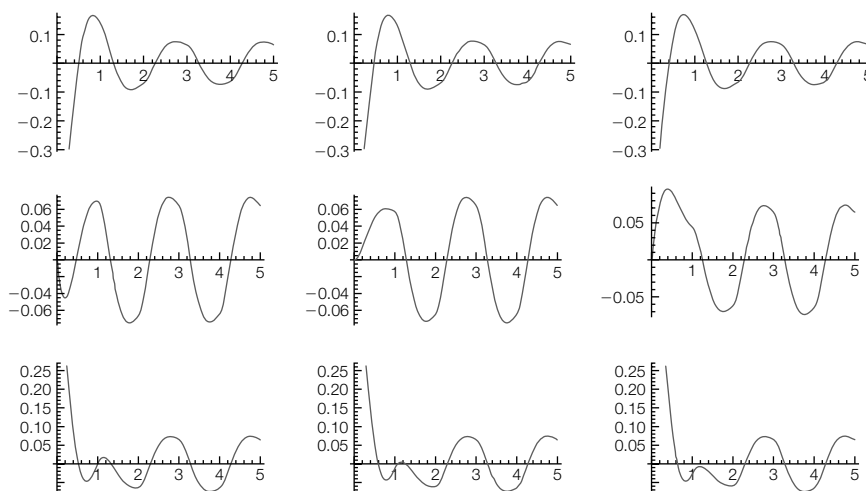
where $g(t) = \frac{1}{20} - \frac{1}{20}e^{-2t} \cos 4t - \frac{1}{40}e^{-2t} \sin 4t$.

To graph the solution for various initial conditions on the interval $[0, 5]$, we define $y_2(t) = g(t) + 2 \sum_{n=1}^5 (-1)^n g(t-n)\mathcal{U}(t-n)$, `sol`, and `inits`. (Note that we can graph the solution for various initial conditions on the interval $[0, m]$ by defining $y_2(t) = g(t) + 2 \sum_{n=1}^m (-1)^n g(t-n)\mathcal{U}(t-n)$.)

```
y2[t_]:=g[t] + 2  $\sum_{n=1}^5 (-1)^n g[t-n]$  UnitStep[t-n]  
Clear[sol]  
sol[t_]:=y1[t] + y2[t]  
inits = {-1/2, 0, 1/2};
```

We then create a table of graphs of `sol[t]` on the interval $[0, 5]$ corresponding to replacing $y(0)$ and $y'(0)$ by the values $-1/2$, 0 , and $1/2$ and then displaying the resulting graphics array in Figure 6.22.

```
graphs = Table[Plot[sol[t]/.{y[0] → inits[[i]], y'[0] → inits[[j]]},  
{t, 0, 5}, DisplayFunction → Identity], {i, 1, 3}, {j, 1, 3}];  
Show[GraphicsGrid[graphs]]
```

**FIGURE 6.22**

Solutions to a differential equation with a piecewise-defined periodic forcing function

Application: The Convolution Theorem

Sometimes we are required to determine the inverse Laplace transform of a product of two functions. Just as in differential and integral calculus when the derivative and integral of a product of two functions did not produce the product of the derivatives and integrals, respectively, neither does the inverse Laplace transform of the product yield the product of the inverse Laplace transforms. *The convolution theorem* tells us how to compute the inverse Laplace transform of a product of two functions.

Theorem 2 (The Convolution Theorem). Suppose that $f(t)$ and $g(t)$ are piecewise continuous on $[0, \infty)$ and both are of exponential order. Furthermore, suppose that the Laplace transform of $f(t)$ is $F(s)$ and that of $g(t)$ is $G(s)$. Then,

$$\mathcal{L}^{-1}\{F(s)G(s)\} = \mathcal{L}^{-1}\{\mathcal{L}\{(f * g)(t)\}\} = \int_0^t f(t - \nu)g(\nu) d\nu. \quad (6.27)$$

Note that $(f * g)(t) = \int_0^t f(t - \nu)g(\nu) d\nu$ is called the **convolution integral**.

Example 6.3.8 (L–R–C Circuits). The initial-value problem used to determine the charge $q(t)$ on the capacitor in an L–R–C circuit is

$$L \frac{d^2 Q}{dt^2} + R \frac{dQ}{dt} + \frac{1}{C} Q = f(t), \quad Q(0) = 0, \quad \frac{dQ}{dt}(0) = 0,$$

where L denotes inductance, $dQ/dt = I$, $I(t)$ current, R resistance, C capacitance, and $E(t)$ voltage supply. Because $dQ/dt = I$, this differential equation can be represented as

$$L \frac{dI}{dt} + RI + \frac{1}{C} \int_0^t I(u) du = E(t).$$

Note also that the initial condition $Q(0) = 0$ is satisfied because $Q(0) = \frac{1}{C} \int_0^0 I(u) du = 0$. The condition $dQ/dt(0) = 0$ is replaced by $I(0) = 0$. (a) Solve this *integro-differential equation*, an equation that involves a derivative as well as an integral of the unknown function, by using the convolution theorem. (b) Consider this example with constant values $L = C = R = 1$ and $E(t) = \begin{cases} \sin t, & 0 \leq t < \pi/2 \\ 0, & t \geq \pi/2 \end{cases}$. Determine $I(t)$ and graph the solution.

Solution We proceed as in the case of a differential equation by taking the Laplace transform of both sides of the equation. The convolution theorem, equation (6.27), is used in determining the Laplace transform of the integral with

$$\mathcal{L} \left\{ \int_0^t I(u) du \right\} = \mathcal{L} \{1 * I(t)\} = \mathcal{L} \{1\} \mathcal{L} \{I(t)\} = \frac{1}{s} \mathcal{L} \{I(t)\}.$$

Therefore, application of the Laplace transform yields

$$Ls\mathcal{L}\{I(t)\} - LI(0) + R\mathcal{L}\{I(t)\} + \frac{1}{C} \frac{1}{s} \mathcal{L}\{I(t)\} = \mathcal{L}\{E(t)\}.$$

Because $I(0) = 0$, we have $Ls\mathcal{L}\{I(t)\} + R\mathcal{L}\{I(t)\} + \frac{1}{C} \frac{1}{s} \mathcal{L}\{I(t)\} = \mathcal{L}\{E(t)\}$. Simplifying and solving for $\mathcal{L}\{I(t)\}$ results in $\mathcal{L}\{I(t)\} = \frac{Cs\mathcal{L}\{E(t)\}}{LCs^2 + RCs + 1}$

Clear[i]

LaplaceTransform[i i'[t] + r i[t], t, s]

r LaplaceTransform[i[t], t, s] +

i(-i[0] + s LaplaceTransform[i[t], t, s])

Solve [1 s lapi + r lapi + $\frac{lapi}{cs} == lape, lapi]$

$\left\{ \left\{ lapi \rightarrow \frac{clapes}{1 + crs + cls^2} \right\} \right\}$

We use lowercase letters to avoid any possible ambiguity with built-in Mathematica functions, such as E and I .

so that $I(t) = \mathcal{L}^{-1} \left\{ \frac{Cs\mathcal{L}\{E(t)\}}{LCs^2 + RCs + 1} \right\}$. In the **Solve** command we use **lapi** to denote $\mathcal{L}\{I(t)\}$ and **lape** to denote $\mathcal{L}\{E(t)\}$. For (b), we note that $E(t) = \begin{cases} \sin t, & 0 \leq t < \pi/2 \\ 0, & t \geq \pi/2 \end{cases}$ can be written as $E(t) = \sin t (\mathcal{U}(t) - \mathcal{U}(t - \pi/2))$. We define and plot the forcing function $E(t)$ on the interval $[0, \pi]$ in Figure 6.23(a).

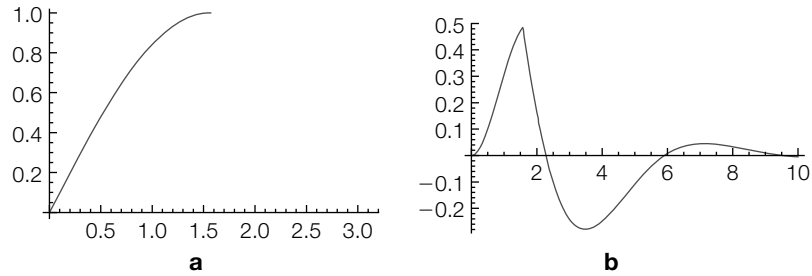


FIGURE 6.23

(a) Plot of $E(t) = \sin t (\mathcal{U}(t) - \mathcal{U}(t - \pi/2))$. (b) $I(t)$ (in black) and $E(t)$ (in gray)

```
e[t_]:=Sin[t] (UnitStep[t] - UnitStep [t -  $\frac{\pi}{2}$ ])
p1 = Plot[e[t], {t, 0,  $\pi$ }]
```

Next, we compute the Laplace transform of $\mathcal{L}\{E(t)\}$ with `LaplaceTransform`. We call this result `lcapi`.

```
lcapi = LaplaceTransform[e[t], t, s]
```

$$\frac{1}{1+s^2} - \frac{e^{-\frac{\pi s}{2}} s}{1+s^2}$$

Using the general formula obtained for the Laplace transform of $I(t)$, we note that the denominator of this expression is given by $s^2 + s + 1$, which is entered as `denom`. Hence, the Laplace transform of $I(t)$, called `lcapi`, is given by the ratio `s lcapi/denom`.

```
denom = s^2 + s + 1;
lcapi = s lcapi/denom;
lcapi = Simplify[lcapi]
```

$$\frac{s - e^{-\frac{\pi s}{2}} s^2}{1 + s + 2s^2 + s^3 + s^4}$$

We determine $I(t)$ with `InverseLaplaceTransform`. Note that `HeavisideTheta[x]` is defined by $\theta(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x > 0 \end{cases}$.

```
i[t_]= InverseLaplaceTransform[lcapi, s, t]
Sin[t] - HeavisideTheta [- $\frac{\pi}{2}$  + t]
( -  $\frac{1}{3} e^{\frac{1}{4}(\pi - 2t)}$  ( 3Cos [ $\frac{1}{4}\sqrt{3}(\pi - 2t)$ ] +  $\sqrt{3}$ Sin [ $\frac{1}{4}\sqrt{3}(\pi - 2t)$ ] ) + Sin[t] ) -
 $\frac{2e^{-t/2}\text{Sin}[\frac{\sqrt{3}t}{2}]}{\sqrt{3}}$  )
```

This solution is plotted in `p2` (in black) and displayed with the forcing function (in gray) in Figure 6.23(b). Notice the effect that the forcing function has on the solution to the differential equation.

```
p2 = Plot[q[t], {t, 0, 10}, DisplayFunction -> Identity];
Show[p1, p2, PlotRange -> All, DisplayFunction -> $DisplayFunction]
Show[GraphicsRow[{p1, p2}]]
```

In this case, we see that we can use `DSolve` to solve the initial-value problem

$$Q'' + Q' + Q = E(t), \quad Q(0) = 0, \quad Q'(0) = 0$$

as well. However, the result is very lengthy, so only a portion is displayed here using `Short`.

```
sol = DSolve[{q''[t] + q'[t] + q[t] == e[t], q[0] == 0, q'[0] == 0}, q[t], t];
Short[sol]
{ { q[t] -> (e^{-t/2} Cos[...]) / (3((1)^2 + (1)^2)) + UnitStep[...], ... } }
```

We see that this result is a real-valued function using `ComplexExpand` followed by `Simplify`.

```
q[t_] = ComplexExpand[sol[[1, 1, 2]]]//Simplify
{ (1/3)e^{-t} sqrt(e^t) (3 Cos[...]) + sqrt(3) (-2 e^{pi/4} Sin[...]) + Sin[...], ... }
{ (1/3)e^{-t} (-3 e^t Cos[t] + sqrt(e^t) (3 Cos[...]) + sqrt(3) Sin[...]) } 0 <= t <= pi/2
```

We use this result to graph $Q(t)$ and $I(t) = Q'(t)$ in Figure 6.24.

```
Plot[{q[t], q'[t]}, {t, 0, 10},
PlotStyle -> {GrayLevel[0], GrayLevel[0.5]}]
```

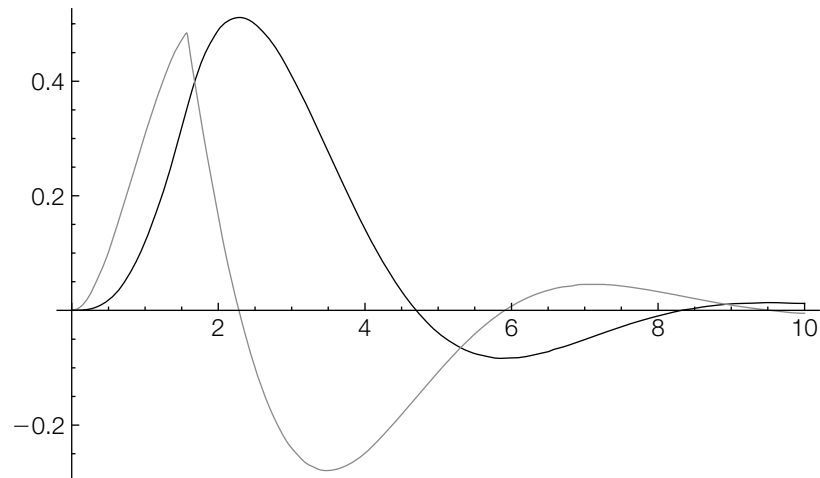


FIGURE 6.24

$Q(t)$ (in black) and $I(t) = Q'(t)$ (in gray)

Application: The Dirac Delta Function

Let $\delta(t - t_0)$ denote the (generalized) function with the two properties

1. $\delta(t - t_0) = 0$ if $t \neq t_0$ and
2. $\int_{-\infty}^{\infty} \delta(t - t_0) dt = 1$

which is called the **Dirac delta function** and is quite useful in the definition of impulse forcing functions that arise in some differential equations. The Laplace transform of $\delta(t - t_0)$ is $\mathcal{L}\{\delta(t - t_0)\} = e^{-st_0}$. The Mathematica function `DiracDelta` represents the δ distribution.

```
LaplaceTransform[DiracDelta[t - t0], t, s]
e-st0HeavisideTheta[t0]
```

Example 6.3.9 Solve $\begin{cases} x'' + x' + x = \delta(t) + \mathcal{U}(t - 2\pi) \\ x(0) = 0, x'(0) = 0 \end{cases}$.

Solution We define `eq` to be the equation $x'' + x' + x = \delta(t) + \mathcal{U}(t - 2\pi)$ and then use `LaplaceTransform` to compute the Laplace transform of `eq`, naming the resulting output `leq`. The symbol `LaplaceTransform[x[t], t, s]` represents the Laplace transform of $x[t]$. We then apply the initial conditions $x(0) = 0$ and $x'(0) = 0$ to `leq` and name the resulting output `ics`.

```
Clear[x, eq]
eq = x''[t] + x'[t] + x[t] == DiracDelta[t] + UnitStep[t - 2π];
leq = LaplaceTransform[eq, t, s]
LaplaceTransform[x[t], t, s] + s LaplaceTransform[x[t], t, s] +
s2 LaplaceTransform[x[t], t, s] - x[0] - sx[0] - x'[0] == 1 +  $\frac{e^{-2\pi s}}{s}$ 
ics = leq /. {x[0] → 0, x'[0] → 0}
LaplaceTransform[x[t], t, s] + s LaplaceTransform[x[t], t, s] +
s2 LaplaceTransform[x[t], t, s] == 1 +  $\frac{e^{-2\pi s}}{s}$ 
```

Next, we use `Solve` to solve the equation `ics` for the Laplace transform of $x(t)$. The expression for the Laplace transform is extracted from `lapx` with `lapx[[1, 1, 2]]`.

```
lapx = Solve[ics, LaplaceTransform[x[t], t, s]]
{ { LaplaceTransform[x[t], t, s] →  $\frac{e^{-2\pi s}(1 + e^{2\pi s}s)}{s(1 + s + s^2)}$  } }
```

To find $x(t)$, we must compute the inverse Laplace transform of the Laplace transform of $\mathcal{L}\{x(t)\}$ obtained in `lapx`. We use `InverseLaplaceTransform` to compute the inverse Laplace transform of `lapx[[1, 1, 2]]` and name the resulting function `x[t]`.

```
x[t_] = InverseLaplaceTransform[lapx[[1, 1, 2]], s, t]
 $\frac{2e^{-t/2}\sin\left[\frac{\sqrt{3}t}{2}\right]}{\sqrt{3}} + \frac{1}{3}\text{HeavisideTheta}[-2\pi + t]$ 
```

$$\left(3 - e^{\pi - \frac{1}{2}} \left(3 \cos \left[\frac{1}{2} \sqrt{3}(-2\pi + t)\right] + \sqrt{3} \sin \left[\frac{1}{2} \sqrt{3}(-2\pi + t)\right]\right)\right)$$

If necessary, to see that this is a real-valued function, we use **ComplexExpand** followed by **Simplify**. If needed, we see that the result is a real-valued function using **ComplexExpand** followed by **Simplify**.

x[t_] = ComplexExpand[x[t]]//Simplify

$$\frac{1}{3}e^{-t} \left(2\sqrt{3}\sqrt{e^t} \sin \left[\frac{\sqrt{3}t}{2}\right] + e^{t/2} \text{HeavisideTheta}[-2\pi + t]\right) \\ \left(3e^{t/2} - 3e^{\pi} \cos \left[\frac{1}{2} \sqrt{3}(-2\pi + t)\right] - \sqrt{3}e^{\pi} \sin \left[\frac{1}{2} \sqrt{3}(-2\pi + t)\right]\right)$$

We use **Plot** to graph the solution on the interval $[0, 8\pi]$ in Figure 6.25.

Plot[x[t], {t, 0, 8\pi}]

Finally, we note that **DSolve** is able to solve the initial-value problem directly as well. The result is very lengthy, so only an abbreviated portion is displayed here using **Short**.

Clear[x]

sol = DSolve[{x''[t] + x'[t] + x[t] == DiracDelta[t] + UnitStep[t - 2\pi],

x[0] == 0, x'[0] == 0}, x[t], t];

Short[sol, 2]

$$\left\{ \left\{ x[t] \rightarrow \frac{e^{-t/2} \langle\langle 1 \rangle\rangle}{3(\cos[\sqrt{3}\pi]^2 + \sin[\langle\langle 1 \rangle\rangle]^2)} + \right. \right. \\ \left. \left(-\frac{2\langle\langle 1 \rangle\rangle \langle\langle 1 \rangle\rangle \sin \left[\frac{\langle\langle 1 \rangle\rangle}{2}\right]}{\sqrt{3}} - \frac{e^{\langle\langle 1 \rangle\rangle} \langle\langle 1 \rangle\rangle}{3\langle\langle 1 \rangle\rangle} \right) \langle\langle 8 \rangle\rangle [\langle\langle 1 \rangle\rangle] \right\} \right\}$$

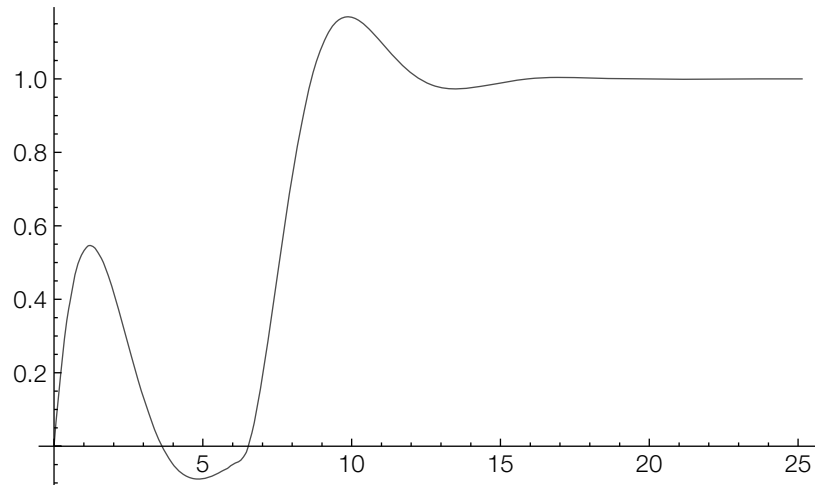


FIGURE 6.25

Plot of $x(t)$ on the interval $[0, 8\pi]$

6.3.5 Nonlinear Higher-Order Equations

Generally, rigorous results regarding nonlinear equations are very difficult to obtain. In some cases, analysis is best carried out numerically and/or graphically. In other situations, rewriting the equation as a system can be of benefit, which is discussed in the next section. (See Examples 6.4.5, 6.4.6, and 6.4.8.)

6.4 SYSTEMS OF EQUATIONS

6.4.1 Linear Systems

We now consider first-order linear systems of differential equations:

$$\mathbf{X}' = \mathbf{A}(t)\mathbf{X} + \mathbf{F}(t), \quad (6.28)$$

where

$$\mathbf{X}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \quad \mathbf{A}(t) = \begin{pmatrix} a_{11}(t) & a_{12}(t) & \cdots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \cdots & a_{2n}(t) \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1}(t) & a_{n2}(t) & \cdots & a_{nn}(t) \end{pmatrix}, \quad \text{and} \quad \mathbf{F}(t) = \begin{pmatrix} f_1(t) \\ f_2(t) \\ \vdots \\ f_n(t) \end{pmatrix}.$$

Homogeneous Linear Systems

The corresponding homogeneous system of equation (6.28) is

$$\mathbf{X}' = \mathbf{A}\mathbf{X}. \quad (6.29)$$

In the same way as with the previously discussed linear equations, a **general solution** of equation (6.28) is $\mathbf{X} = \mathbf{X}_b + \mathbf{X}_p$, where $\mathbf{X}_b$ is a *general solution* of equation (6.29) and $\mathbf{X}_p$ is a *particular solution* of the nonhomogeneous system equation (6.28).

If $\Phi_1, \Phi_2, \dots, \Phi_n$ are n linearly independent solutions of equation (6.29), a **general solution** of equation (6.29) is

$$\mathbf{X} = c_1\Phi_1 + c_2\Phi_2 + \cdots + c_n\Phi_n = (\Phi_1 \ \Phi_2 \ \cdots \ \Phi_n) \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \Phi \mathbf{C},$$

where

$$\Phi = (\Phi_1 \ \Phi_2 \ \cdots \ \Phi_n) \quad \text{and} \quad \mathbf{C} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}.$$

A **particular solution** to a system of ordinary differential equations is a set of functions that satisfy the system but do not contain any arbitrary constants. That is, a particular solution to a system is a set of specific functions, containing no arbitrary constants, that satisfy the system.

Φ is called a **fundamental matrix** for equation (6.29). If Φ is a fundamental matrix for equation (6.29), $\Phi' = A\Phi$ or $\Phi' - A\Phi = 0$.

$A(t)$ constant

Suppose that $A(t) = A$ has constant real entries. Let λ be an eigenvalue of A with corresponding eigenvector $\mathbf{v}$. Then, $\mathbf{v}e^{\lambda t}$ is a solution of $\mathbf{X}' = A\mathbf{X}$.

If $\lambda = \alpha + \beta i$, $\beta \neq 0$, is an eigenvalue of A and has corresponding eigenvector $\mathbf{v} = \mathbf{a} + \beta i\mathbf{b}$, two linearly independent solutions of $\mathbf{X}' = A\mathbf{X}$ are

$$e^{\alpha t} (\mathbf{a} \cos \beta t - \mathbf{b} \sin \beta t) \quad \text{and} \quad e^{\alpha t} (\mathbf{a} \sin \beta t + \mathbf{b} \cos \beta t). \quad (6.30)$$

Example 6.4.1 Solve each of the following systems:

$$(a) \mathbf{X}' = \begin{pmatrix} -1/2 & -1/3 \\ -1/3 & -1/2 \end{pmatrix} \mathbf{X}; \quad (b) \begin{cases} x' = \frac{1}{2}y \\ y' = -\frac{1}{8}x \end{cases}; \quad (c) \begin{cases} dx/dt = -\frac{1}{4}x + 2y \\ dy/dt = -8x - \frac{1}{4}y. \end{cases}$$

Solution (a) With Eigensystem, we see that the eigenvalues and eigenvectors of $A = \begin{pmatrix} -1/2 & -1/3 \\ -1/3 & -1/2 \end{pmatrix}$ are $\lambda_1 = -1/6$ and $\lambda_2 = -5/6$ and $\mathbf{v}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ and $\mathbf{v}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, respectively.

capa = {{ -1/2, -1/3}, {-1/3, -1/2}};

Eigensystem[capa]

{{{-5/6, -1/6}, {{1, 1}, {-1, 1}}}}

Then $\mathbf{X}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} e^{-t/6}$ and $\mathbf{X}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} e^{-5t/6}$ are two linearly independent solutions

of the system, so a general solution is $\mathbf{X} = \begin{pmatrix} -e^{-t/6} & e^{-5t/6} \\ e^{-t/6} & e^{-5t/6} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$; a fundamental

matrix is $\Phi = \begin{pmatrix} -e^{-t/6} & e^{-5t/6} \\ e^{-t/6} & e^{-5t/6} \end{pmatrix}$.

We use DSolve to find a general solution of the system by entering

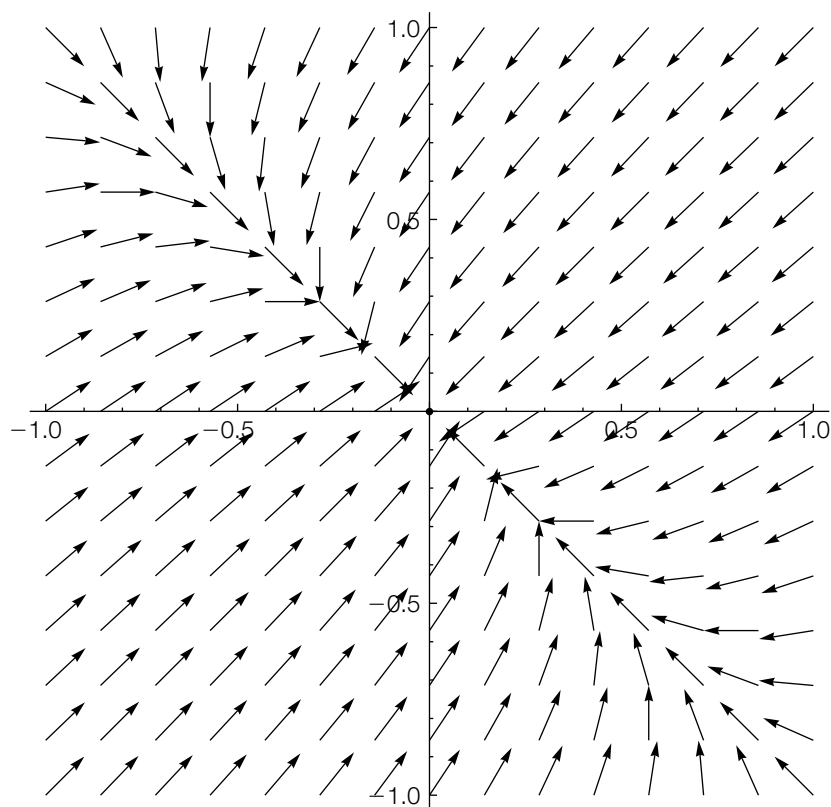
Clear[x, y]

gensol = DSolve[{x'[t] == -1/2x[t] - 1/3y[t],

y'[t] == -1/3x[t] - 1/2y[t]}, {x[t], y[t]}, t]

{{x[t] -> 1/2 e^{-5t/6} (1 + e^{2t/3}) C[1] - 1/2 e^{-5t/6} (-1 + e^{2t/3}) C[2],
y[t] -> -1/2 e^{-5t/6} (-1 + e^{2t/3}) C[1] + 1/2 e^{-5t/6} (1 + e^{2t/3}) C[2]}}

We graph the direction field with VectorFieldPlot, which is contained in the VectorFieldPlots, in Figure 6.26.

**FIGURE 6.26**Direction field for $\mathbf{X}' = \mathbf{A}\mathbf{X}$

Remark 6.4 After you have loaded the `VectorFieldPlots` package,

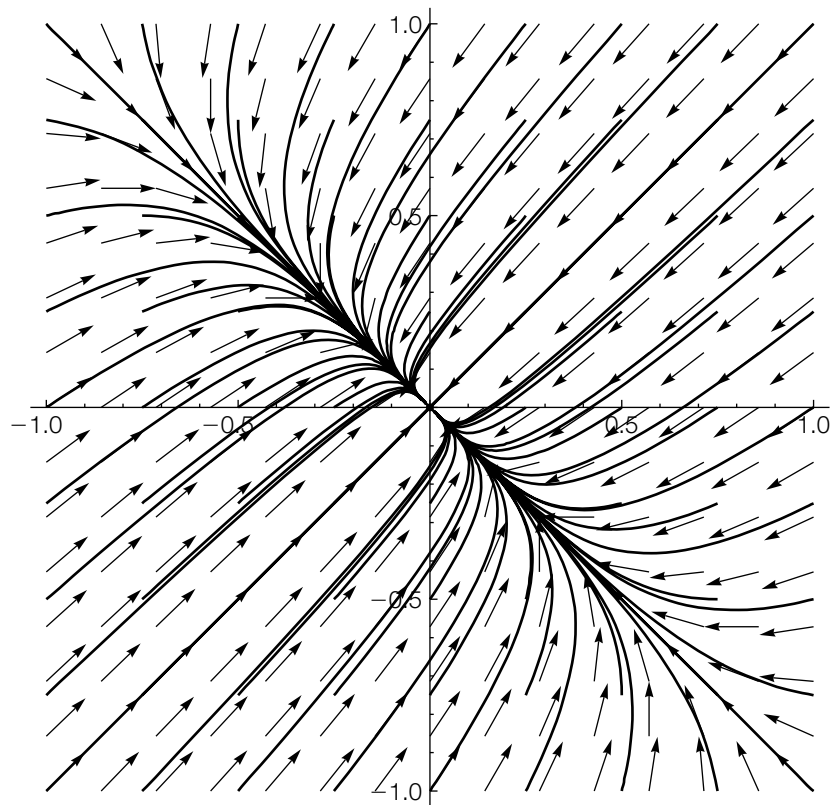
`VectorFieldPlot[{f[x, y], g[x, y]}, {x, a, b}, {y, c, d}]`

generates a basic direction field for the system $\{x' = f(x, y), y' = g(x, y)\}$ for $a \leq x \leq b$ and $c \leq y \leq d$.

`<< "VectorFieldPlots"`

**`pvf = Show[VectorFieldPlot[{-1/2x - 1/3y, -1/3x - 1/2y},
{x, -1, 1}, {y, -1, 1}, ScaleFunction -> (1&)],
Axes -> Automatic]`**

Several solutions are also graphed with `ParametricPlot` and shown together with the direction field in Figure 6.27. To do so, we first solve the system if $x(0) = x_0$ and $y(0) = y_0$.

**FIGURE 6.27**

Direction field for $\mathbf{X}' = \mathbf{A}\mathbf{X}$ along with various solution curves

```

initsol = DSolve[{x'[t]== -1/2x[t] - 1/3y[t],
  y'[t]== -1/3x[t] - 1/2y[t], x[0]==x0, y[0]==y0}, {x[t], y[t]}, t]
{ {x[t] → ½ e-5t/6 (x0 + e2t/3x0 + y0 - e2t/3y0),
   y[t] → -½ e-5t/6 (-x0 + e2t/3x0 - y0 - e2t/3y0) } }

```

Given an ordered pair, **solplot** parametrically graphs the solution satisfying $x(0) = x_0$ and $y(0) = y_0$ for $0 \leq t \leq 15$.

```

solplot[pair_]:=
ParametricPlot[
  Evaluate[{x[t], y[t]}/.initsol/.{x0 → pair[[1]], y0 → pair[[2]]},
    {t, 0, 15}, PlotStyle → {{Black, Thickness[.005]}}

```

We then define a list of ordered pairs with **Table** followed by **Flatten**

```
Clear[i, j]
orderedpairs = Flatten[Table[{i, j}, {i, -1, 1, 1/4}, {j, -1, 1, 1/4}], 1];
Short[orderedpairs]
 $\{\{-1, -1\}, \{-1, -\frac{3}{4}\}, \{-1, -\frac{1}{2}\}, \langle\langle 75 \rangle\rangle, \{1, \frac{1}{2}\}, \{1, \frac{3}{4}\}, \{1, 1\}\}$ 
```

and use **Map** to apply **solplot** to **orderedpairs**.

```
toshow = Map[solplot, orderedpairs];
```

The resulting list of graphics objects is displayed together with **Show**. See Figure 6.27.

```
Show[toshow, pvf, PlotRange → {{-1, 1}, {-1, 1}}]
```

(b) In matrix form the system is equivalent to the system $\mathbf{X}' = \begin{pmatrix} 0 & 1/2 \\ -1/8 & 0 \end{pmatrix} \mathbf{X}$.

As in (a), we use **Eigensystem** to see that the eigenvalues and eigenvectors of $\mathbf{A} = \begin{pmatrix} 0 & 1/2 \\ -1/8 & 0 \end{pmatrix}$ are $\lambda_{1,2} = 0 \pm \frac{1}{4}i$ and $\mathbf{v}_{1,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \pm \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} i$.

```
capa = {{0, 1/2}, {-1/8, 0}};
Eigensystem[capa]
 $\{\{\frac{i}{4}, -\frac{i}{4}\}, \{\{-2i, 1\}, \{2i, 1\}\}\}$ 
```

Two linearly independent solutions are then $\mathbf{X}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cos \frac{1}{4}t - \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} \sin \frac{1}{4}t = \begin{pmatrix} \cos \frac{1}{4}t \\ -\frac{1}{2} \sin \frac{1}{4}t \end{pmatrix}$ and $\mathbf{X}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \sin \frac{1}{4}t + \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} \cos \frac{1}{4}t = \begin{pmatrix} \sin \frac{1}{4}t \\ \frac{1}{2} \cos \frac{1}{4}t \end{pmatrix}$, and a general solution is $\mathbf{X} = c_1 \mathbf{X}_1 + c_2 \mathbf{X}_2 = \begin{pmatrix} \cos \frac{1}{4}t & \sin \frac{1}{4}t \\ -\frac{1}{2} \sin \frac{1}{4}t & \frac{1}{2} \cos \frac{1}{4}t \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$ or $x = c_1 \cos \frac{1}{4}t + c_2 \sin \frac{1}{4}t$ and $y = -c_1 \frac{1}{2} \sin \frac{1}{4}t + \frac{1}{2} c_2 \cos \frac{1}{4}t$.

As before, we use **DSolve** to find a general solution.

```
Clear[x, y]
gensol = DSolve[{x'[t]==1/2y[t], y'[t]==-1/8x[t]},
{x[t], y[t]}, t]
 $\{\{x[t] \rightarrow C[1] \cos \left[\frac{1}{4}t\right] + 2C[2] \sin \left[\frac{1}{4}t\right],$ 
 $y[t] \rightarrow C[2] \cos \left[\frac{1}{4}t\right] - \frac{1}{2}C[1] \sin \left[\frac{1}{4}t\right]\}\}$ 
```

Initial-value problems for systems are solved in the same way as for other equations. For example, entering

```
partsol = DSolve[{x'[t]==1/2y[t], y'[t]==-1/8x[t],
```

```

x[0]==1, y[0]==-1},
{x[t], y[t]}, t]
{ {x[t] → Cos [1/4]-2Sin [1/4], y[t] → 1/2 (-2Cos [1/4]-Sin [1/4]) } }

```

finds the solution that satisfies $x(0) = 1$ and $y(0) = -1$.

We graph $x(t)$ and $y(t)$ together as well as parametrically with `Plot` and `ParametricPlot`, respectively, in Figure 6.28.

```

p1 = Plot[{x[t], y[t]}/.partsol, {t, 0, 8Pi}];
p2 = ParametricPlot[{x[t], y[t]}/.partsol, {t, 0, 8Pi},
  AspectRatio → Automatic];
Show[GraphicsRow[{p1, p2}]]

```

We can also use `VectorFieldPlot` and `ParametricPlot` to graph the direction field and/or various solutions as we do next with `Manipulate`. In this case, `Manipulate` is used to graph the solution passing through the locator point. As you move the point with the cursor, the corresponding solution is plotted. See Figure 6.29.

```

Clear[pvf, sol, p1];
Manipulate[
  << "VectorFieldPlots";
  Module[{pvf, sol, p1},
    pvf = Show[VectorFieldPlot[{1/2y, -1/8x},
      {x, -2, 2}, {y, -1, 1}, ScaleFunction → {1&}],
      Axes → Automatic];
    sol = DSolve[{x'[t]==1/2y[t], y'[t]==-1/8x[t],
      x[0]==pt[[1]], y[0]==pt[[2]]},
      {x[t], y[t]}, t];
    p1 = ParametricPlot[{x[t], y[t]}/.sol, {t, 0, 8Pi},
      PlotStyle → Thickness[.01]];
    Show[p1, pvf, PlotRange → {{-2, 2}, {-1, 1}},
      AspectRatio → 1], {{pt, {1, .5}}, Locator}]

```

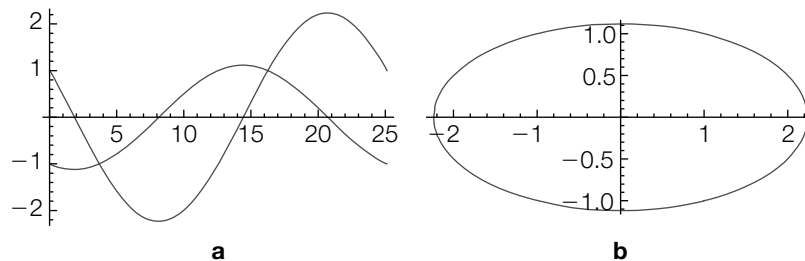
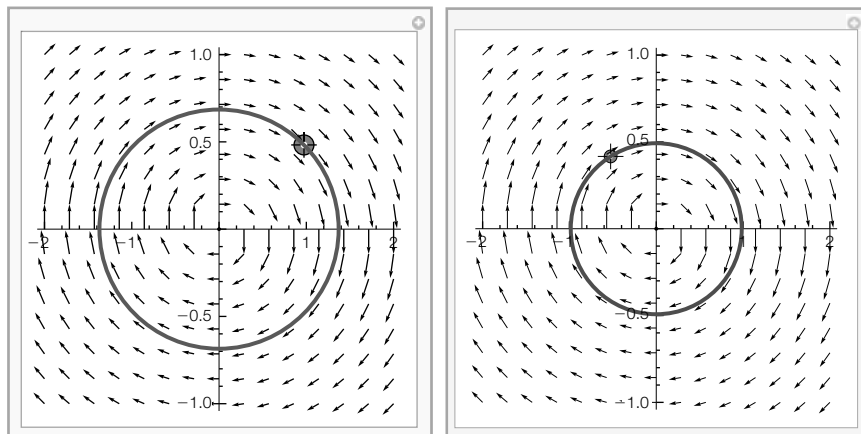


FIGURE 6.28

(a) Graph of $x(t)$ and $y(t)$. (b) Parametric plot of $x(t)$ versus $y(t)$

**FIGURE 6.29**

With Manipulate, notice that all nontrivial solutions are periodic

(c) In matrix form, the system is equivalent to the system $\mathbf{X}' = \begin{pmatrix} -\frac{1}{4} & 2 \\ -8 & -\frac{1}{4} \end{pmatrix} \mathbf{X}$.

The eigenvalues and corresponding eigenvectors of $\mathbf{A} = \begin{pmatrix} -\frac{1}{4} & 2 \\ -8 & -\frac{1}{4} \end{pmatrix}$ are found to be $\lambda_{1,2} = -\frac{1}{4} \pm 4i$ and $\mathbf{v}_{1,2} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \pm \begin{pmatrix} 1 \\ 0 \end{pmatrix} i$ with Eigensystem.

```
capa = {{-1/4, 2}, {-8, -1/4}};
```

```
Eigensystem[capa]
```

```
{{{-1/4 + 4i, -1/4 - 4i}, {{-i/2, 1}, {i/2, 1}}}}
```

A general solution is then

$$\begin{aligned} \mathbf{X} &= c_1 \mathbf{X}_1 + c_2 \mathbf{X}_2 \\ &= c_1 e^{-t/4} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \cos 4t - \begin{pmatrix} 0 \\ 2 \end{pmatrix} \sin 4t \right) + c_2 e^{-t/4} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \sin 4t + \begin{pmatrix} 0 \\ 2 \end{pmatrix} \cos 4t \right) \\ &= e^{-t/4} \left[c_1 \begin{pmatrix} \cos 4t \\ -2 \sin 4t \end{pmatrix} + c_2 \begin{pmatrix} \sin 4t \\ 2 \cos 4t \end{pmatrix} \right] = e^{-t/4} \begin{pmatrix} \cos 4t & \sin 4t \\ -2 \sin 4t & 2 \cos 4t \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \end{aligned}$$

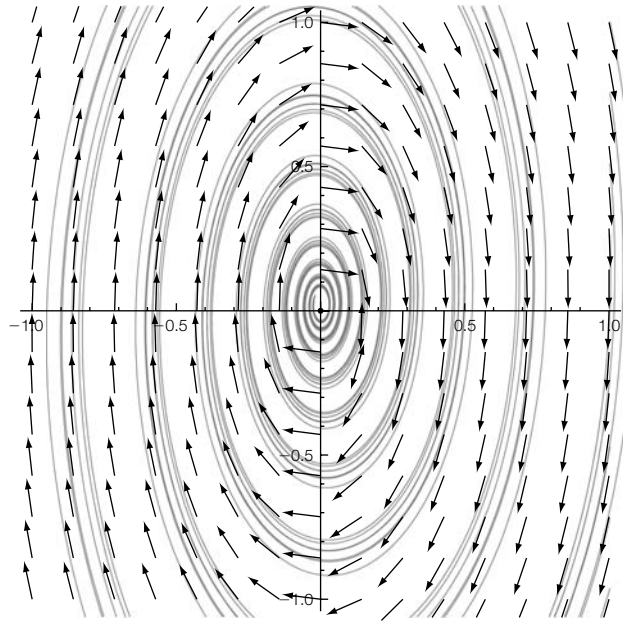
or $x = e^{-t/4} (c_1 \cos 4t + c_2 \sin 4t)$ and $y = e^{-t/4} (2c_2 \cos 4t - 2c_1 \sin 4t)$. We confirm this result using DSolve.

```
gensol = DSolve[{x'[t] == -1/4 x[t] + 2 y[t],
```

```
y'[t] == -8 x[t] - 1/4 y[t]}, {x[t], y[t]}, t]
```

```
{ {x[t] -> e^{-t/4} C[1] Cos[4t] + 1/2 e^{-t/4} C[2] Sin[4t],
```

```
y[t] -> e^{-t/4} C[2] Cos[4t] - 2 e^{-t/4} C[1] Sin[4t] } }
```

**FIGURE 6.30**

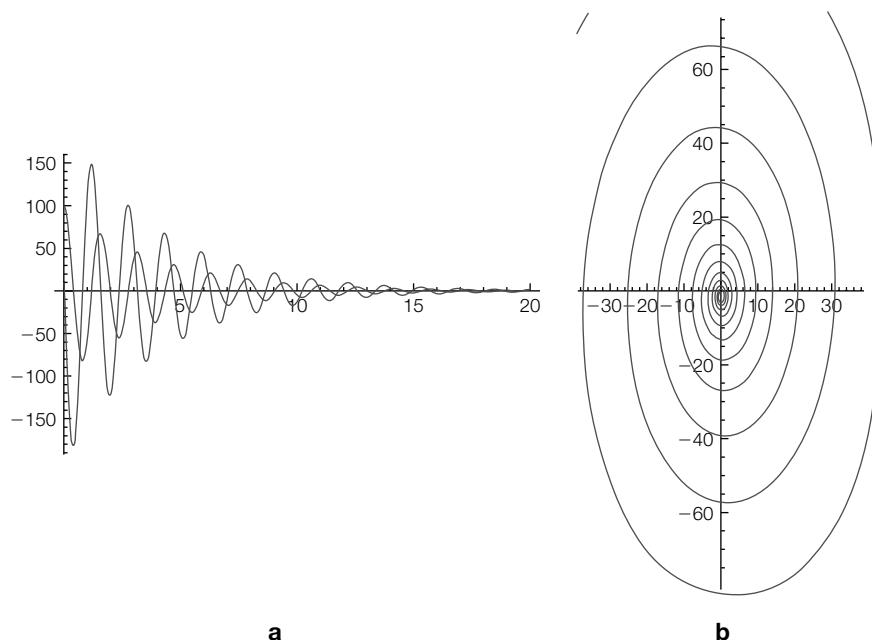
Various solutions and direction field associated with the system

We use `VectorFieldPlot` and `ParametricPlot` to graph the direction field associated with the system along with various solutions in Figure 6.30.

```
Clear[x, y]
initsol = DSolve[{x'[t] == -1/4 x[t] + 2 y[t],
  y'[t] == -8 x[t] - 1/4 y[t], x[0] == x0, y[0] == y0},
  {x[t], y[t]}, t]
{ {x[t] -> 1/2 e^{-t/4} (2 x0 Cos[4t] + y0 Sin[4t]),
  y[t] -> e^{-t/4} (y0 Cos[4t] - 2 x0 Sin[4t]) } }

t1 = Table[ParametricPlot[{x[t], y[t]}/.initsol/.
  {x0 -> 1, y0 -> i}, {t, 0, 15}, PlotStyle -> Gray],
  {i, -1, 1, 2/8}];
pvf = Show[VectorFieldPlot[{-1/4 x + 2 y, -8 x - 1/4 y},
  {x, -1, 1}, {y, -1, 1}, ScaleFunction -> (1 &)],
  Axes -> Automatic];
Show[t1, pvf, PlotRange -> {{-1, 1}, {-1, 1}},
  AspectRatio -> Automatic]
```

Last, we illustrate how to solve an initial-value problem and graph the resulting solutions by finding the solution that satisfies the initial conditions $x(0) = 100$ and $y(0) = 10$ and then graphing the results with `Plot` and `ParametricPlot` in Figure 6.31.

**FIGURE 6.31**

(a) Graph of $x(t)$ and $y(t)$. (b) Parametric plot of $x(t)$ versus $y(t)$ (for help with `Show` and `GraphicsRow` use the **Documentation Center**)

```
Clear[x, y]
partsol = DSolve[{x'[t] == -1/4 x[t] + 2 y[t],
  y'[t] == -8 x[t] - 1/4 y[t], x[0] == 100, y[0] == 10},
  {x[t], y[t]}, t]

{ {x[t] -> 5 e^{-t/4} (20 Cos[4t] + Sin[4t]), y[t] -> 10 e^{-t/4} (Cos[4t] - 20 Sin[4t]) } }

p1 = Plot[{x[t], y[t]}/.partsol, {t, 0, 20}, PlotRange -> All];
p2 = ParametricPlot[{x[t], y[t]}/.partsol, {t, 0, 20},
  AspectRatio -> Automatic];
Show[GraphicsRow[{p1, p2}]]
```

Application: The Double Pendulum

The motion of a double pendulum is modeled by the system of differential equations

$$\begin{cases} (m_1 + m_2) l_1^2 \frac{d^2 \theta_1}{dt^2} + m_2 l_1 l_2 \frac{d^2 \theta_2}{dt^2} + (m_1 + m_2) l_1 g \theta_1 = 0 \\ m_2 l_2^2 \frac{d^2 \theta_2}{dt^2} + m_2 l_1 l_2 \frac{d^2 \theta_1}{dt^2} + m_2 l_2 g \theta_2 = 0 \end{cases}$$

using the approximation $\sin \theta \approx \theta$ for small displacements. θ_1 represents the displacement of the upper pendulum and θ_2 that of the lower pendulum. Also, m_1 and m_2 represent the mass attached to the upper and lower pendulums, respectively, whereas the length of each is given by l_1 and l_2 .

Example 6.4.2 Suppose that $m_1 = 3$, $m_2 = 1$, and each pendulum has length 16. If $\theta_1(0) = 1$, $\theta_1'(0) = 0$, $\theta_2(0) = -1$, and $\theta_2'(0) = 0$, solve the double pendulum problem using $g = 32$. Plot the solution.

Solution In this case, the system to be solved is

$$\begin{cases} 4 \cdot 16^2 \frac{d^2 \theta_1}{dt^2} + 16^2 \frac{d^2 \theta_2}{dt^2} + 4 \cdot 16 \cdot 32 \theta_1 = 0 \\ 16^2 \frac{d^2 \theta_2}{dt^2} + 16^2 \frac{d^2 \theta_1}{dt^2} + 16 \cdot 32 \theta_2 = 0, \end{cases}$$

which we simplify to obtain

$$\begin{cases} 4 \frac{d^2 \theta_1}{dt^2} + \frac{d^2 \theta_2}{dt^2} + 8 \theta_1 = 0 \\ \frac{d^2 \theta_2}{dt^2} + \frac{d^2 \theta_1}{dt^2} + 2 \theta_2 = 0. \end{cases}$$

In the following code, we let $x(t)$ and $y(t)$ represent $\theta_1(t)$ and $\theta_2(t)$, respectively. First, we use `DSolve` to solve the initial-value problem.

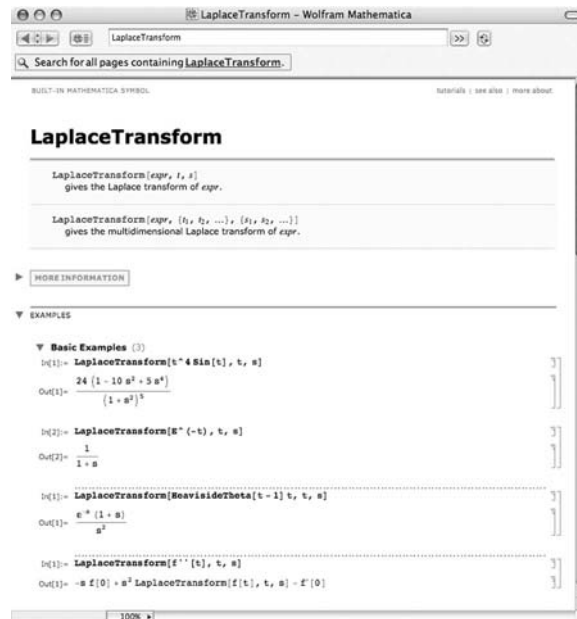
```
sol = DSolve[{4x''[t] + y''[t] + 8x[t] == 0, x''[t] + y''[t] + 2y[t] == 0,
  x[0] == 1, x'[0] == 1, y[0] == 0, y'[0] == -1}, {x[t], y[t]}, t]
{ {x[t] -> 1/8 (4Cos[2t] + 4Cos[2t/sqrt[3]] + 3Sin[2t] + sqrt[3]Sin[2t/sqrt[3]]),
  y[t] -> 1/4 (-4Cos[2t] + 4Cos[2t/sqrt[3]] - 3Sin[2t] + sqrt[3]Sin[2t/sqrt[3]]) } }
```

To solve the initial-value problem using traditional methods, we use the *method of Laplace transforms*. To do so, we define `sys` to be the system of equations and use `LaplaceTransform` to compute the Laplace transform of each equation.

step1 = LaplaceTransform[sys, t, s]

```
{8LaplaceTransform[x[t], t, s] +
s^2LaplaceTransform[y[t], t, s] - sy[0] +
4 (s^2LaplaceTransform[x[t], t, s] - sx[0] - x'[0]) - y'[0] == 0,
s^2LaplaceTransform[x[t], t, s] +
2LaplaceTransform[y[t], t, s] +
s^2LaplaceTransform[y[t], t, s] - sy[0] -
sy[0] - x'[0] - y'[0] == 0}
```

The Laplace transform of $y = f(t)$ is $F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty e^{-st} f(t) dt$.



Next, we apply the initial conditions and solve the resulting system of equations for $\mathcal{L}\{\theta_1(t)\} = X(s)$ and $\mathcal{L}\{\theta_2(t)\} = Y(s)$.

step2 = step1/.{x[0] -> 1, x'[0] -> 1, y[0] -> 0, y'[0] -> -1}

$$\begin{aligned} & \{1 + 8\text{LaplaceTransform}[x[t], t, s] + \\ & 4(-1 - s + s^2\text{LaplaceTransform}[x[t], t, s]) + \\ & s^2\text{LaplaceTransform}[y[t], t, s] == 0, \\ & -s + s^2\text{LaplaceTransform}[x[t], t, s] + \\ & 2\text{LaplaceTransform}[y[t], t, s] + \\ & s^2\text{LaplaceTransform}[y[t], t, s] == 0\} \end{aligned}$$

step3 =

Solve[step2, {LaplaceTransform[x[t], t, s],
LaplaceTransform[y[t], t, s]]

$$\left\{ \left\{ \text{LaplaceTransform}[x[t], t, s] \rightarrow -\frac{-6 - 8s - 3s^2 - 3s^3}{16 + 16s^2 + 3s^4}, \right. \right. \\ \left. \left. \text{LaplaceTransform}[y[t], t, s] \rightarrow -\frac{-8s + 3s^2}{16 + 16s^2 + 3s^4} \right\} \right\}$$

InverseLaplaceTransform is then used to find $\theta_1(t)$ and $\theta_2(t)$.

$f(t)$ is the **inverse Laplace transform** of $F(s)$ if $\mathcal{L}\{f(t)\} = F(s)$; we write $\mathcal{L}^{-1}\{F(s)\} = f(t)$.

$$\begin{aligned} \mathbf{x[t_]} &= \text{InverseLaplaceTransform} \left[-\frac{-6 - 8s - 3s^2 - 3s^3}{16 + 16s^2 + 3s^4}, \mathbf{s, t} \right] \\ &= \frac{1}{8} \left(4\text{Cos}[2t] + 4\text{Cos} \left[\frac{2t}{\sqrt{3}} \right] + 3\text{Sin}[2t] + \sqrt{3}\text{Sin} \left[\frac{2t}{\sqrt{3}} \right] \right) \end{aligned}$$

$$y[t_]=\text{InverseLaplaceTransform}\left[-\frac{-8s+3s^2}{16+16s^2+3s^4},s,t\right]$$

$$\frac{1}{4}\left(-4\cos[2t]+4\cos\left[\frac{2t}{\sqrt{3}}\right]-3\sin[2t]+\sqrt{3}\sin\left[\frac{2t}{\sqrt{3}}\right]\right)$$

These two functions are graphed together in Figure 6.32(a) and parametrically in Figure 6.32(b).

```
p1 = Plot[{x[t], y[t]}, {t, 0, 20},
  PlotStyle -> {GrayLevel[0], GrayLevel[0.5]}]
p2 = ParametricPlot[{x[t], y[t]}, {t, 0, 20},
  PlotRange -> {{-5/2, 5/2}, {-5/2, 5/2}}, AspectRatio -> 1]
Show[GraphicsRow[{p1, p2}]]
```

We can illustrate the motion of the pendulum as follows. First, we define the function pen2.

```
Clear[pen2]
pen2[t_, len1_, len2_] := Module[{pt1, pt2},
  pt1 = {len1 Cos [3π/2 + x[t]], len1 Sin [3π/2 + x[t]]};
  pt2 = {len1 Cos [3π/2 + x[t]] + len2 Cos [3π/2 + y[t]],
    len1 Sin [3π/2 + x[t]] + len2 Sin [3π/2 + y[t]]};
  Show[Graphics[{Line[{{0, 0}, pt1}], PointSize[.05], Point[pt1],
    Line[{{pt1, pt2}], PointSize[.05], Point[pt2]}], Axes -> Automatic,
    Ticks -> None, AxesStyle -> GrayLevel[.5],
    PlotRange -> {{-32, 32}, {-34, 0}},
    DisplayFunction -> Identity]]
```

Next, we define tvals to be a list of 16 evenly spaced numbers between 0 and 10. Map is then used to apply pen2 to the list of numbers in tvals. The resulting

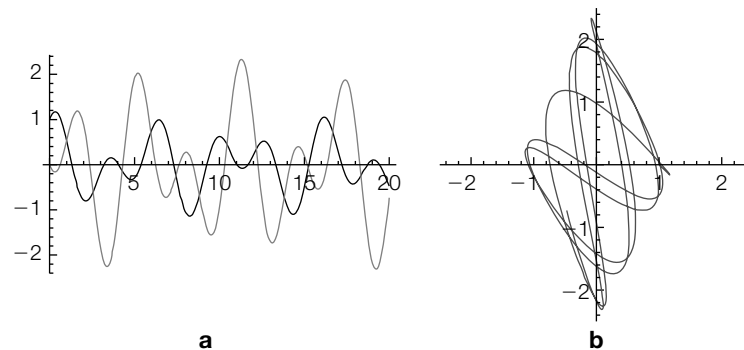
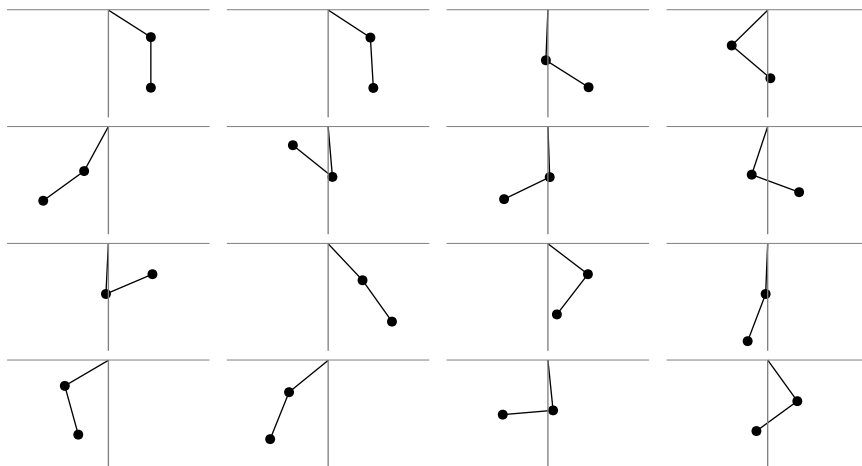


FIGURE 6.32

(a) $\theta_1(t)$ (in black) and $\theta_2(t)$ (in gray) as functions of t . (b) Parametric plot of $\theta_1(t)$ versus $\theta_2(t)$

**FIGURE 6.33**

The double pendulum for 16 equally spaced values of t between 0 and 10

set of graphics is partitioned into four element subsets and displayed using `Show` and `GraphicsGrid` in Figure 6.33.

```
tvals = Table [t, {t, 0, 10,  $\frac{10}{16}$ }] ;
graphs = Map[pen2[#, 16, 16]&, tvals];
toshow = Partition[graphs, 4];
Show[GraphicsGrid[toshow]]
```

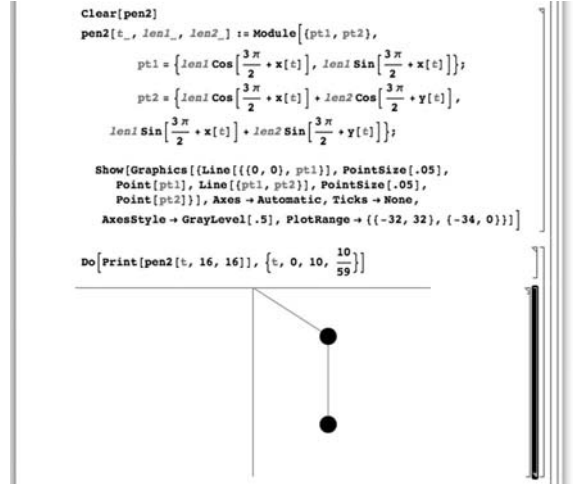
If the option `DisplayFunction->Identity` is omitted from the definition of `pen2`, we can use a `Do` loop together with `Print` to generate a set of graphics that can then be animated.

```
Clear[pen2]
pen2[t_, len1_, len2_] := Module[{pt1, pt2},
pt1 = { len1 Cos [ $\frac{3\pi}{2} + x[t]$ ] , len1 Sin [ $\frac{3\pi}{2} + x[t]$ ] };
pt2 = { len1 Cos [ $\frac{3\pi}{2} + x[t]$ ] + len2 Cos [ $\frac{3\pi}{2} + y[t]$ ] ,
len1 Sin [ $\frac{3\pi}{2} + x[t]$ ] + len2 Sin [ $\frac{3\pi}{2} + y[t]$ ] };
Show[Graphics[{Line[{0, 0}, pt1]}, PointSize[.05], Point[pt1],
Line[{pt1, pt2}], PointSize[.05], Point[pt2]}], Axes -> Automatic,
Ticks -> None, AxesStyle -> GrayLevel[.5],
PlotRange -> {{-32, 32}, {-34, 0}}]]
```

We show one frame from the animation that results from the `Do` loop

```
Do [Print[pen2[t, 16, 16]], {t, 0, 10,  $\frac{10}{59}$ }]
```

in the following screen shot.



Alternatively, you can use Manipulate

```
Clear[pen2]
Manipulate[

$$x[t_] = \frac{1}{8} \left( 4\cos[2t] + 4\cos\left[\frac{2t}{\sqrt{3}}\right] + 3\sin[2t] + \sqrt{3}\sin\left[\frac{2t}{\sqrt{3}}\right] \right);$$

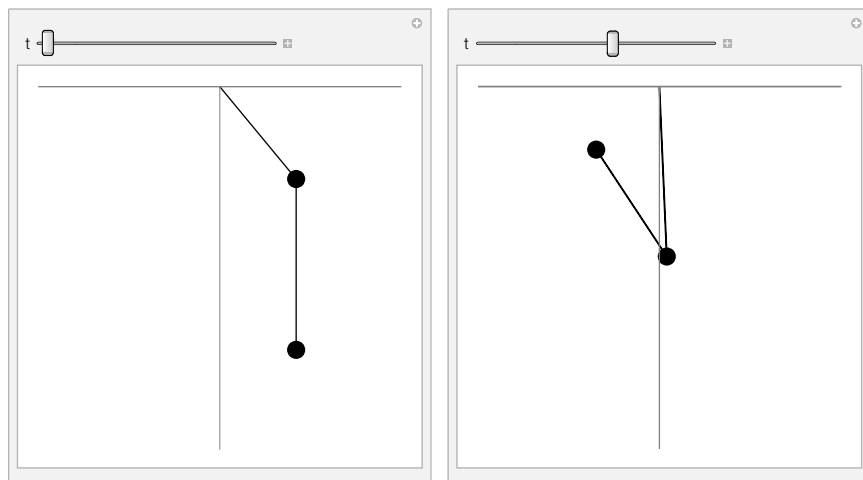

$$y[t_] = \frac{1}{4} \left( -4\cos[2t] + 4\cos\left[\frac{2t}{\sqrt{3}}\right] - 3\sin[2t] + \sqrt{3}\sin\left[\frac{2t}{\sqrt{3}}\right] \right);$$

pen2[t_, len1_, len2_] := Module[{pt1, pt2},
  pt1 = {len1 Cos[ $\frac{3\pi}{2} + x[t]$ ], len1 Sin[ $\frac{3\pi}{2} + x[t]$ ]};
  pt2 = {len1 Cos[ $\frac{3\pi}{2} + x[t]$ ] + len2 Cos[ $\frac{3\pi}{2} + y[t]$ ],
    len1 Sin[ $\frac{3\pi}{2} + x[t]$ ] + len2 Sin[ $\frac{3\pi}{2} + y[t]$ ]};
  Show[Graphics[{Line[{{0, 0}, pt1}], PointSize[.05],
    Point[pt1], Line[{pt1, pt2}], PointSize[.05],
    Point[pt2]}], Axes → Automatic, Ticks → None,
    AxesStyle → GrayLevel[.5], AspectRatio → 1,
    PlotRange → {{-32, 32}, {-34, 0}}];
  pen2[t, 16, 16], {t, 0, 10,  $\frac{10}{59}$ }]
```

to generate a nearly identical animation as shown in Figure 6.34.

6.4.2 Nonhomogeneous Linear Systems

Generally, the method of undetermined coefficients is difficult to implement for nonhomogeneous linear systems because the choice for the particular solution must be very carefully made. Variation of parameters is implemented in much the same way as for first-order linear equations.

**FIGURE 6.34**

Two frames from a Manipulate animation of a double pendulum

Let $\mathbf{X}_h$ be a general solution to the corresponding homogeneous system of equation (6.28), $\mathbf{X}$ a general solution of equation (6.28), and $\mathbf{X}_p$ a particular solution of equation (6.28). It then follows that $\mathbf{X} - \mathbf{X}_p$ is a solution to the corresponding homogeneous system so $\mathbf{X} - \mathbf{X}_p = \mathbf{X}_h$ and, consequently, $\mathbf{X} = \mathbf{X}_h + \mathbf{X}_p$. A particular solution of equation (6.28) is found in much the same way as with first-order linear equations. Let Φ be a fundamental matrix for the corresponding homogeneous system. We assume that a particular solution has the form $\mathbf{X}_p = \Phi \mathbf{U}(t)$. Differentiating $\mathbf{X}_p$ gives us

$$\mathbf{X}_p' = \Phi' \mathbf{U} + \Phi \mathbf{U}'.$$

Substituting into equation (6.28) results in

$$\Phi' \mathbf{U} + \Phi \mathbf{U}' = \mathbf{A} \Phi \mathbf{U} + \mathbf{F}$$

$$\Phi \mathbf{U}' = \mathbf{F}$$

$$\mathbf{U}' = \Phi^{-1} \mathbf{F}$$

$$\mathbf{U} = \int \Phi^{-1} \mathbf{F} dt,$$

where we have used the fact that $\Phi' \mathbf{U} - \mathbf{A} \Phi \mathbf{U} = (\Phi' - \mathbf{A} \Phi) \mathbf{U} = \mathbf{0}$. It follows that

$$\mathbf{X}_p = \Phi \int \Phi^{-1} \mathbf{F} dt. \quad (6.31)$$

A general solution is then

$$\begin{aligned} \mathbf{X} &= \mathbf{X}_h + \mathbf{X}_p \\ &= \Phi \mathbf{C} + \Phi \int \Phi^{-1} \mathbf{F} dt \end{aligned}$$

$$= \Phi \left(C + \int \Phi^{-1} \mathbf{F} dt \right) = \Phi \int \Phi^{-1} \mathbf{F} dt,$$

where we have incorporated the constant vector $\mathbf{C}$ into the indefinite integral $\int \Phi^{-1} \mathbf{F} dt$.

Example 6.4.3 Solve the initial-value problem

$$\mathbf{X}' = \begin{pmatrix} 1 & -1 \\ 10 & -1 \end{pmatrix} \mathbf{X} - \begin{pmatrix} t \cos 3t \\ t \sin t + t \cos 3t \end{pmatrix}, \quad \mathbf{X}(0) = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Remark 6.5 In traditional form, the system is equivalent to

$$\begin{cases} x' = x - y - t \cos 3t \\ y' = 10x - y - t \sin t - t \cos 3t, \end{cases} \quad x(0) = 1, y(0) = -1.$$

Solution The corresponding homogeneous system is $\mathbf{X}'_h = \begin{pmatrix} 1 & -1 \\ 10 & -1 \end{pmatrix} \mathbf{X}_h$. The eigenvalues and corresponding eigenvectors of $\mathbf{A} = \begin{pmatrix} 1 & -1 \\ 10 & -1 \end{pmatrix}$ are $\lambda_{1,2} = \pm 3i$ and $\mathbf{v}_{1,2} = \begin{pmatrix} 1 \\ 10 \end{pmatrix} \pm \begin{pmatrix} -3 \\ 0 \end{pmatrix} i$, respectively.

capa = {{1, -1}, {10, -1}};

Eigensystem[capa]

$\{\{3i, -3i\}, \{\frac{1}{10} + \frac{3i}{10}, 1\}, \{\frac{1}{10} - \frac{3i}{10}, 1\}\}$

A fundamental matrix is $\Phi = \begin{pmatrix} \sin 3t & \cos 3t \\ \sin 3t - 3 \cos 3t & \cos 3t + 3 \sin 3t \end{pmatrix}$ with inverse $\Phi^{-1} = \begin{pmatrix} \frac{1}{3} \cos 3t + \sin 3t & -\frac{1}{3} \cos 3t \\ -\frac{1}{3} \sin 3t + \cos 3t & \frac{1}{3} \sin 3t \end{pmatrix}$.

fm = {{Sin[3t], Sin[3t] - 3Cos[3t]}, {Cos[3t], Cos[3t] + 3Sin[3t]}};

fminv = Inverse[fm]//Simplify

$\{\{\frac{1}{3} \cos 3t + \sin 3t, \cos 3t - \frac{1}{3} \sin 3t\}, \{-\frac{1}{3} \cos 3t, \frac{1}{3} \sin 3t\}\}$

We now compute $\Phi^{-1} \mathbf{F}(t)$

ft = {-tCos[3t], -tSin[t] - tCos[3t]};

step1 = fminv.ft

$\{(-t \cos 3t - t \sin t)(\cos 3t - \frac{1}{3} \sin 3t) -$
 $t \cos 3t (\frac{1}{3} \cos 3t + \sin 3t),$
 $\frac{1}{3} t \cos 3t^2 + \frac{1}{3} (-t \cos 3t - t \sin t) \sin 3t\}$

and $\int \Phi^{-1} \mathbf{F}(t) dt$.

step2 = Integrate[step1, t]

$$\begin{aligned} & \left\{ -\frac{t^2}{3} + \frac{1}{24}\cos[2t] - \frac{1}{4}t\cos[2t] - \frac{1}{96}\cos[4t] + \frac{1}{8}t\cos[4t] - \right. \\ & \quad \frac{1}{54}\cos[6t] + \frac{1}{18}t\cos[6t] + \frac{1}{8}\sin[2t] + \frac{1}{12}t\sin[2t] - \\ & \quad \frac{1}{32}\sin[4t] - \frac{1}{24}t\sin[4t] - \frac{1}{108}\sin[6t] - \frac{1}{9}t\sin[6t], \\ & \quad \frac{t^2}{12} - \frac{1}{24}\cos[2t] + \frac{1}{96}\cos[4t] + \frac{1}{216}\cos[6t] + \frac{1}{36}t\cos[6t] - \\ & \quad \left. \frac{1}{12}t\sin[2t] + \frac{1}{24}t\sin[4t] - \frac{1}{216}\sin[6t] + \frac{1}{36}t\sin[6t] \right\} \end{aligned}$$

A general solution of the nonhomogeneous system is then $\Phi \left(\int \Phi^{-1} \mathbf{F}(t) dt + \mathbf{C} \right)$.

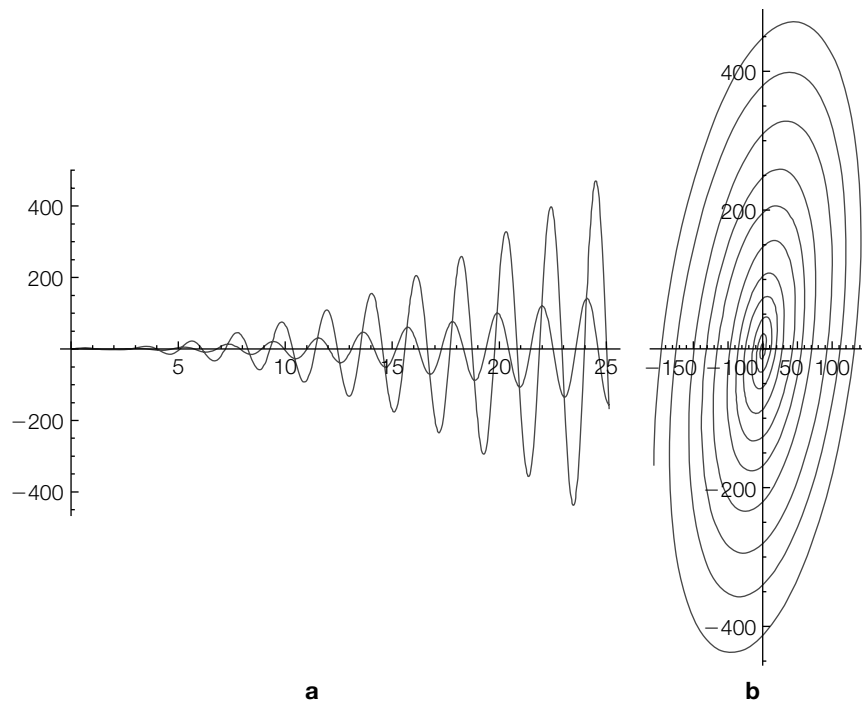
Simplify[fm . step2]

$$\begin{aligned} & \left\{ \frac{1}{288}(27\cos[t] - \right. \\ & \quad 4((1 + 6t + 18t^2)\cos[3t] + 27t\sin[t] + (-1 + 6t + 18t^2)\sin[3t]), \\ & \quad \frac{1}{288}(-36t\cos[t] - 4(1 - 6t + 18t^2)\cos[3t] - \\ & \quad \left. 45\sin[t] - 4\sin[3t] - 24t\sin[3t] + 72t^2\sin[3t]) \right\} \end{aligned}$$

It is easiest to use **DSolve** to solve the initial-value problem directly as we do next.

**check = DSolve[{x'[t]==x[t]-y[t]-tCos[3t], y'[t]==
10x[t]-y[t]-tSin[t]-tCos[3t], x[0]==1, y[0]==-1},
{x[t], y[t]}, t]**

$$\begin{aligned} & \left\{ \left\{ x[t] \rightarrow \frac{1}{288} \right. \right. \\ & \quad (301\cos[3t] - 72t^2\cos[3t] - 12\cos[2t]\cos[3t] \\ & \quad + 3\cos[3t]\cos[4t] - 4\cos[3t]\cos[6t] - 24t\cos[3t]\sin[2t] \\ & \quad + 192\sin[3t] + 24t\cos[2t]\sin[3t] - 12t\cos[4t]\sin[3t] \\ & \quad + 24t\cos[6t]\sin[3t] - 12\sin[2t]\sin[3t] + 12t\cos[3t]\sin[4t] \\ & \quad + 3\sin[3t]\sin[4t] - 24t\cos[3t]\sin[6t] - 4\sin[3t]\sin[6t]), \\ & \quad y[t] \rightarrow \frac{1}{288}(-275\cos[3t] - 72t^2\cos[3t] \\ & \quad - 12\cos[2t]\cos[3t] - 72t\cos[2t]\cos[3t] \\ & \quad + 3\cos[3t]\cos[4t] + 36t\cos[3t]\cos[4t] - 4\cos[3t]\cos[6t] \\ & \quad - 72t\cos[3t]\cos[6t] + 36\cos[3t]\sin[2t] - 24t\cos[3t]\sin[2t] \\ & \quad + 1095\sin[3t] - 216t^2\sin[3t] - 36\cos[2t]\sin[3t] \\ & \quad + 24t\cos[2t]\sin[3t] + 9\cos[4t]\sin[3t] - 12t\cos[4t]\sin[3t] \\ & \quad - 12\cos[6t]\sin[3t] + 24t\cos[6t]\sin[3t] - 12\sin[2t]\sin[3t] \\ & \quad - 72t\sin[2t]\sin[3t] - 9\cos[3t]\sin[4t] + 12t\cos[3t]\sin[4t] \\ & \quad + 3\sin[3t]\sin[4t] + 36t\sin[3t]\sin[4t] + 12\cos[3t]\sin[6t] \\ & \quad \left. \left. - 24t\cos[3t]\sin[6t] - 4\sin[3t]\sin[6t] - 72t\sin[3t]\sin[6t]) \right\} \right\} \end{aligned}$$

**FIGURE 6.35**

(a) Graph of $x(t)$ (in black) and $y(t)$ (in gray). (b) Parametric plot of $x(t)$ versus $y(t)$

The solutions are graphed with `Plot` and `ParametricPlot` in Figure 6.35.

```
p1 = Plot[{x[t], y[t]}/.check, {t, 0, 8π}, PlotRange → All];
p2 = ParametricPlot[Evaluate[{x[t], y[t]}/.check], {t, 0, 8π},
  AspectRatio → Automatic];
Show[GraphicsRow[{p1, p2}]]
```

In the case that $\mathbf{A}$ is constant, $\mathbf{X}' = \mathbf{A}\mathbf{X}$ is called an *autonomous system* and the only *equilibrium (rest point)* solution is the zero solution: $\mathbf{X} = \mathbf{0}$. The stability of the solution is determined by the eigenvalues of $\mathbf{A}$. If all the eigenvalues of $\mathbf{A}$ have negative real part, then $\mathbf{X} = \mathbf{0}$ is **globally asymptotically stable** because $\lim_{t \rightarrow \infty} \mathbf{X}(t) = \mathbf{0}$ for *all* solutions. If not all the eigenvalues of $\mathbf{A}$ have negative real part, then $\mathbf{X} = \mathbf{0}$ is unstable.

For the 2×2 system, $\mathbf{X}' = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mathbf{X}$ or, equivalently, $x' = ax + by$, $y' = cx + dy$, the stability of $(0, 0)$ is easily seen by examining the direction field

for the system. If all vectors lead to the origin, it is stable; if they do not, it is not.

Example 6.4.4 The eigenvalues of $\begin{pmatrix} -\alpha & \beta \\ -\beta & 0 \end{pmatrix}$ are $\lambda_{1,2} = \frac{1}{2}(-a \pm \sqrt{a^2 - 4b^2})$. (See the exercises.)

Thus, $(0, 0)$ is globally asymptotically stable for the system $\mathbf{X}' = \begin{pmatrix} -\alpha & \beta \\ -\beta & 0 \end{pmatrix} \mathbf{X}$.

With **Manipulate**, you can investigate the various situations. In the following, we can vary α and β and then plot the solution passing through each locator point. Several results are shown in Figure 6.36.

```
Manipulate[
Needs["VectorFieldPlots`"];
sol1 = DSolve[{x'[t] == -αx[t] + βy[t], y'[t] == -βx[t],
  x[0] == init1[[1]], y[0] == init1[[2]]}, {x[t], y[t]}, t];
sol2 = DSolve[{x'[t] == -αx[t] + βy[t], y'[t] == -βx[t], x[0] == init2[[1]],
  y[0] == init2[[2]]}, {x[t], y[t]}, t];
psol1 = ParametricPlot[{x[t], y[t]}/.sol1, {t, -20, 20},
  PlotStyle → {{GrayLevel[.3], Thickness[.01]}}, PlotPoints → 200];
psol2 = ParametricPlot[{x[t], y[t]}/.sol2, {t, -20, 20},
  PlotStyle → {{GrayLevel[.6], Thickness[.01]}}, PlotPoints → 200];
p1 = Show[VectorFieldPlot[{-αx + βy, -βx}, {x, -1, 1}, {y, -1, 1},
  ScaleFunction → (1 &)], psol1, psol2],
Axes → Automatic, AxesOrigin → {0, 0}, PlotRange → {{-1, 1}, {-1, 1}},
  AspectRatio → Automatic,
{{α, 1}, -2.5, 5}, {{β, 2}, -2.5, 5}, {{init1, {.5, .5}}, Locator},
  {{init2, {- .5, - .5}}, Locator}]
```

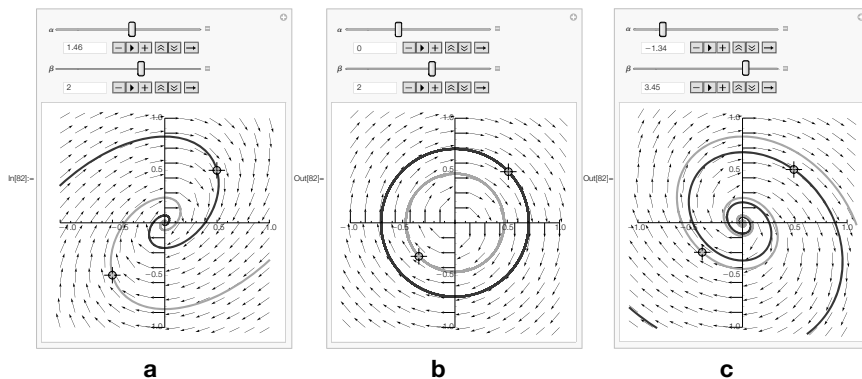


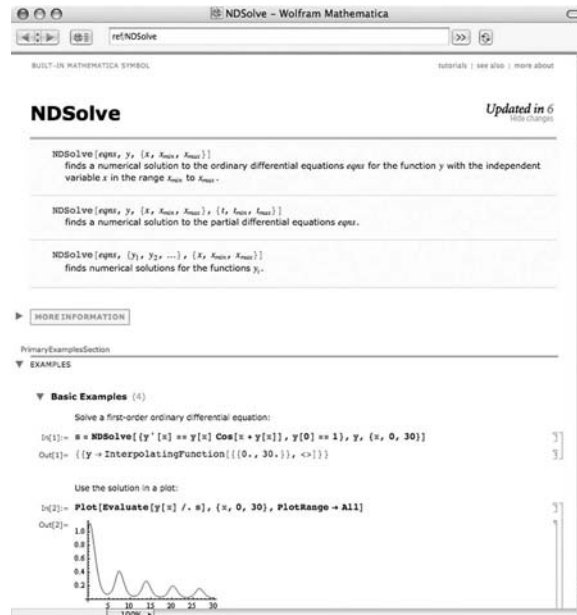
FIGURE 6.36

As we vary α and β and change the initial conditions, the system behaves differently. (a) A stable spiral. (b) A center. (c) An unstable spiral

6.4.3 Nonlinear Systems

Nonlinear systems of differential equations arise in numerous situations. Rigorous analysis of the behavior of solutions to nonlinear systems is usually very difficult, if not impossible.

To generate numerical solutions of equations, use `NDSolve`.



Example 6.4.5 (Van der Pol's equation). Van der Pol's equation, $x'' + \mu(x^2 - 1)x' + x = 0$

Also see Example 6.4.8.

can be written as the system

$$\begin{aligned} x' &= y \\ y' &= -x - \mu(x^2 - 1)y. \end{aligned} \quad (6.32)$$

If $\mu = 2/3$, $x(0) = 1$, and $y(0) = 0$, (a) find $x(1)$ and $y(1)$. (b) Graph the solution that satisfies these initial conditions.

Solution We use `NDSolve` together to solve equation (6.32) with $\mu = 2/3$ subject to $x(0) = 1$ and $y(0) = 0$. We name the resulting numerical solution `numsol`.

```
numsol = NDSolve[{x'[t]==y[t], y'[t]==-x[t]-2/3(x[t]^2-1)y[t], x[0]==1,
y[0]==0}, {x[t], y[t]}, {t, 0, 30}]
{{x[t] -> InterpolatingFunction[{{0., 30.}}, <>][t],
```

We evaluate `numsol` if $t = 1$ to see that $x(1) \approx 0.5128$ and $y(1) \approx -0.9692$.

```
y[t] → InterpolatingFunction[{{0., 30.}}, <>][t]]}
```

```
{x[t], y[t]}/.numsol/.t->1
```

```
{{0.512848, -0.969204}}
```

`Plot`, `ParametricPlot`, and `ParametricPlot3D` are used to graph $x(t)$ and $y(t)$ together in Figure 6.37(a); a three-dimensional plot, $(t, x(t), y(t))$, is shown in Figure 6.37(b); a parametric plot is shown in Figure 6.37(c); and the limit

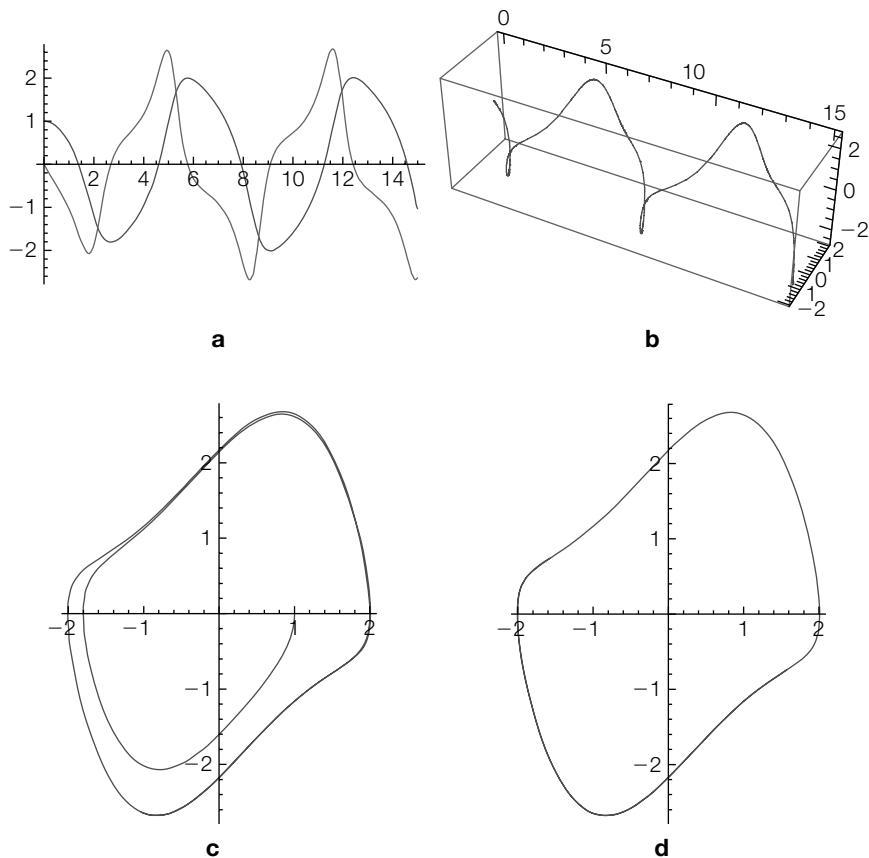


FIGURE 6.37

(a) $x(t)$ and $y(t)$. (b) A three-dimensional plot. (c) $x(t)$ versus $y(t)$. (d) $x(t)$ versus $y(t)$ for $20 \leq t \leq 30$

cycle is shown more clearly in Figure 6.37(d) by graphing the solution for $20 \leq t \leq 30$.

```
p1 = Plot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 15}];
p2 = ParametricPlot3D[Evaluate[{t, x[t], y[t]}/.numsol], {t, 0, 15}];
p3 = ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 15},
  AspectRatio → Automatic];
p4 = ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 20, 30},
  AspectRatio → Automatic];
Show[GraphicsGrid[{{p1, p2}, {p3, p4}}]]
```

To consider other μ values, decide on a μ range, combine the previous commands, replace $2/3$ with μ , and use `Manipulate`. See Figure 6.38.

To avoid conflicts with the variables in the `Manipulate`, consider quitting Mathematica, restarting, and then entering the `Manipulate` command in a new notebook.

```
Manipulate[
  numsol = NDSolve[{x'[t]==y[t], y'[t]==-x[t]-μ(x[t]^2-1)y[t],
    x[0]==1, y[0]==0}, {x[t], y[t]}, {t, 0, 30}];
  p1 = Plot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 15},
    PlotRange → All, AspectRatio → 1];
  p2 = ParametricPlot3D[Evaluate[{t, x[t], y[t]}/.numsol], {t, 0, 15},
    BoxRatios → {4, 1, 1}];
  p3 = ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 15},
    AspectRatio → 1, PlotRange → All];
  p4 = ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 20, 30},
    AspectRatio → 1, PlotRange → All];
  Show[GraphicsGrid[{{p1, p2}, {p3, p4}}], {{μ, 1}, 0, 3}]
```

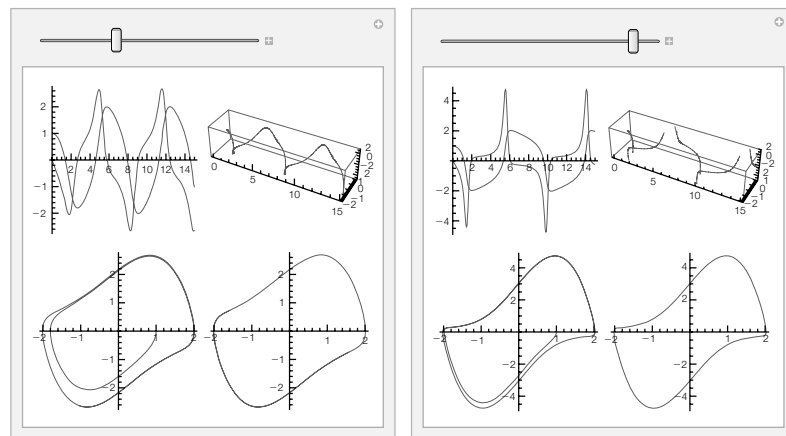


FIGURE 6.38

Plots of solutions of Van der Pol's equation for various values of μ

Linearization

An **autonomous system** does not explicitly depend on the independent variable, t . That is, if you write the system omitting all arguments, the independent variable (typically t) does not appear.

Consider the autonomous system of the form

$$\begin{aligned}x_1' &= f_1(x_1, x_2, \dots, x_n) \\x_2' &= f_2(x_1, x_2, \dots, x_n) \\&\vdots \\x_n' &= f_n(x_1, x_2, \dots, x_n).\end{aligned}\tag{6.33}$$

An **equilibrium** (or **rest**) **point**, $E = (x_1^*, x_2^*, \dots, x_n^*)$, of equation (6.33) is a solution of the system

$$\begin{aligned}f_1(x_1, x_2, \dots, x_n) &= 0 \\f_2(x_1, x_2, \dots, x_n) &= 0 \\&\vdots \\f_n(x_1, x_2, \dots, x_n) &= 0.\end{aligned}\tag{6.34}$$

The **Jacobian** of equation (6.33) is

$$\mathbf{J}(x_1, x_2, \dots, x_n) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}.$$

The rest point, E , is **locally stable** if and only if all the eigenvalues of $\mathbf{J}(E)$ have negative real part. If E is not locally stable, E is **unstable**.

Example 6.4.6 (Duffing's Equation). Consider the forced **pendulum equation** with damping,

Establishing global stability of an equilibrium point for a nonlinear system is *significantly* more difficult than establishing global stability of an equilibrium point ($E = (0, 0)$) for a linear autonomous system.

$$x'' + kx' + \omega \sin x = F(t).\tag{6.35}$$

Recall the Maclaurin series for $\sin x$: $\sin x = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \dots$. Using $\sin x \approx x$, equation (6.35) reduces to the linear equation $x'' + kx' + \omega x = F(t)$.

On the other hand, using the approximation $\sin x \approx x - \frac{1}{6}x^3$, we obtain $x'' + kx' + \omega(x - \frac{1}{6}x^3) = F(t)$. Adjusting the coefficients of x and x^3 and assuming that $F(t) = F \cos \omega t$ gives us **Duffing's equation**:

$$x'' + kx' + cx + \epsilon x^3 = F \cos \omega t,\tag{6.36}$$

where k and c are positive constants.

Let $y = x'$. Then, $y' = x'' = F \cos \omega t - kx' - cx - \epsilon x^3 = F \cos \omega t - ky - cx - \epsilon x^3$ and we can write equation (6.36) as the system

$$\begin{aligned} x' &= y \\ y' &= F \cos \omega t - ky - cx - \epsilon x^3. \end{aligned} \quad (6.37)$$

Assuming that $F = 0$ results in the autonomous system

$$\begin{aligned} x' &= y \\ y' &= -cx - \epsilon x^3 - ky. \end{aligned} \quad (6.38)$$

The rest points of system equation (6.38) are found by solving

$$\begin{aligned} x' &= 0 \\ y' &= -cx - \epsilon x^3 - ky = 0, \end{aligned}$$

resulting in $E_0 = (0, 0)$.

Solve[{y==0, -cx-εx^3-ky==0}, {x, y}]

$$\left\{ \{y \rightarrow 0, x \rightarrow 0\}, \left\{ y \rightarrow 0, x \rightarrow -\frac{i\sqrt{c}}{\sqrt{\epsilon}} \right\}, \left\{ y \rightarrow 0, x \rightarrow \frac{i\sqrt{c}}{\sqrt{\epsilon}} \right\} \right\}$$

We find the Jacobian of equation (6.38) in s1, evaluate the Jacobian at E_0 ,

s1 = {{0, 1}, {-c-3εx^2, -k}};

s2 = s1/.x->0

$$\left\{ \{0, 1\}, \{-c, -k\} \right\}$$

and then compute the eigenvalues with **Eigenvalues**.

s3 = Eigenvalues[s2]

$$\left\{ \frac{1}{2} \left(-k - \sqrt{-4c + k^2} \right), \frac{1}{2} \left(-k + \sqrt{-4c + k^2} \right) \right\}$$

Because k and c are positive, $k^2 - 4c < k^2$, so the real part of each eigenvalue is always negative if $k^2 - 4c \neq 0$. Thus, E_0 is locally stable.

For the autonomous system

$$\begin{aligned} x' &= f(x, y) \\ y' &= g(x, y), \end{aligned} \quad (6.39)$$

Bendixson's theorem states that if $f_x(x, y) + g_y(x, y)$ is a continuous function that is either always positive or always negative in a particular region R of the plane, then system (6.39) has no limit cycles in R . For equation (6.38), we have

$$\frac{d}{dx}(y) + \frac{d}{dy}(-cx - \epsilon x^3 - ky) = -k,$$

which is always negative. Hence, equation (6.38) has no limit cycles and it follows that E_0 is globally, asymptotically stable.

$$D[y, x] + D[-cx - \epsilon x^3 - ky, y]$$

−k

We use `VectorFieldPlot` and `ParametricPlot` to illustrate two situations that occur. In Figure 6.39(a), we use $c = 1$, $\epsilon = 1/2$, and $k = 3$. In this case, E_0 is a *stable node*. On the other hand, in Figure 6.39(b), we use $c = 10$, $\epsilon = 1/2$, and $k = 3$. In this case, E_0 is a *stable spiral*.

```
Needs["VectorFieldPlots"];
pvf1 = VectorFieldPlots`VectorFieldPlot [ { y, -x -  $\frac{x^3}{2}$  - 3y }, {x, -2.5, 2.5 },
      {y, -2.5, 2.5}];
numgraph[init_, c_, opts___]:=Module[{numsol},
  numsol = NDSolve[{x'[t]==y[t], y'[t]== -cx[t] - 1/2x[t]^3 - 3y[t],
    x[0]==init[[1]], y[0]==init[[2]]}, {x[t], y[t]}, {t, 0, 10}];
  ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 10},
    opts, DisplayFunction->Identity]]
i1 = Table[numgraph[{2.5, i}, 1], {i, -2.5, 2.5, 1/2}];
i2 = Table[numgraph[{-2.5, i}, 1], {i, -2.5, 2.5, 1/2}];
i3 = Table[numgraph[{i, 2.5}, 1], {i, -2.5, 2.5, 1/2}];
i4 = Table[numgraph[{i, -2.5}, 1], {i, -2.5, 2.5, 1/2}];
c1 = Show[i1, i2, i3, i4, pvf1, PlotRange -> {{-2.5, 2.5}, {-2.5, 2.5}},
  AspectRatio -> Automatic]

pvf2 = VectorFieldPlots`VectorFieldPlot [ { y, -10x -  $\frac{x^3}{2}$  - 3y }, {x, -2.5, 2.5 },
      {y, -2.5, 2.5}];
i1 = Table[numgraph[{2.5, i}, 10], {i, -2.5, 2.5, 1/2}];
```

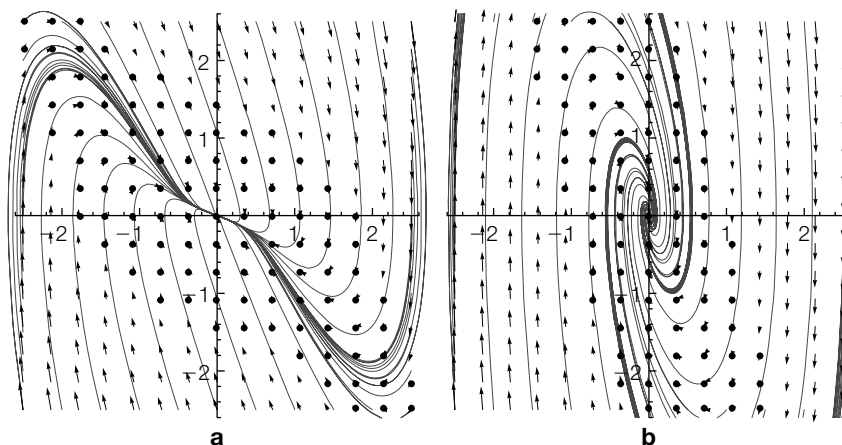


FIGURE 6.39

(a) The origin is a stable node. (b) The origin is a stable spiral

```

i2 = Table[numgraph[{-2.5, i}, 10], {i, -2.5, 2.5, 1/2}];
i3 = Table[numgraph[{i, 2.5}, 10], {i, -2.5, 2.5, 1/2}];
i4 = Table[numgraph[{i, -2.5}, 10], {i, -2.5, 2.5, 1/2}];
c2 = Show[i1, i2, i3, i4, pvf2, PlotRange -> {{-2.5, 2.5}, {-2.5, 2.5}},
  AspectRatio -> Automatic]
Show[GraphicsRow[{c1, c2}]]

```

To experiment with different parameter values, use `Manipulate`. In the following, we investigate how varying c from 0 to 10 affects the solutions of Duffing's equation. See Figure 6.40.

```

Clear[pvf, i1, i2, i3, i4];
Manipulate[
Needs["VectorFieldPlots"];
numgraph[init_, c_, opts_____]:=Module[{numsol},
numsol = NDSolve[{x'[t]==y[t], y'[t]==-cx[t]-1/2x[t]^3-3y[t],
  x[0]==init[[1]], y[0]==init[[2]]}, {x[t], y[t]}, {t, 0, 10}];
ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 10},
  opts, DisplayFunction->Identity]];
pvf = VectorFieldPlots`VectorFieldPlot[ $\left\{y, -cx - \frac{x^3}{2} - 3y\right\}$ , {x, -2.5, 2.5},
  {y, -2.5, 2.5}];
i1 = Table[numgraph[{2.5, i}, c], {i, -2.5, 2.5, 1/2}];
i2 = Table[numgraph[{-2.5, i}, c], {i, -2.5, 2.5, 1/2}];
i3 = Table[numgraph[{i, 2.5}, c], {i, -2.5, 2.5, 1/2}];
i4 = Table[numgraph[{i, -2.5}, c], {i, -2.5, 2.5, 1/2}];

```

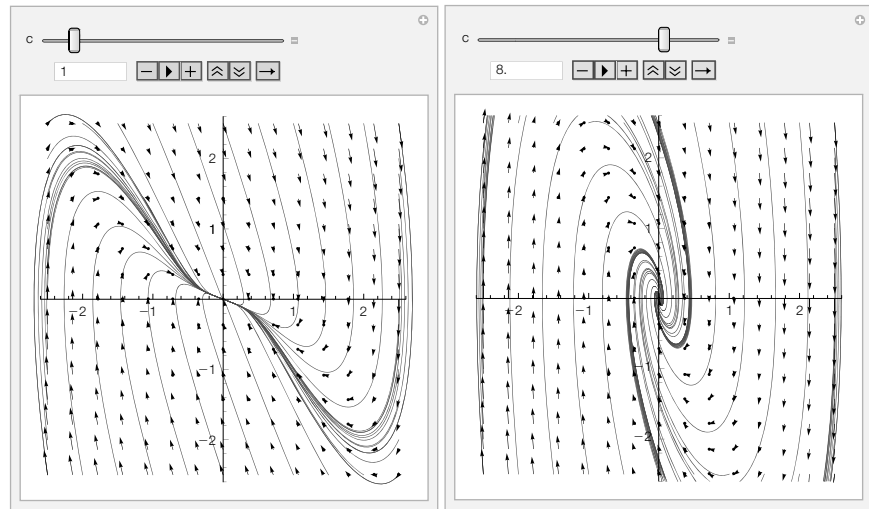


FIGURE 6.40

Allowing c to vary in Duffing's equation

Show[i1, i2, i3, i4, pvf, PlotRange → {{-2.5, 2.5}, {-2.5, 2.5}},
AspectRatio → Automatic], {{c, 1}, 0, 10}]

Example 6.4.7 (Predator–Prey). One form of the **predator–prey** is

There are *many*
 other predator–prey
 models.

$$\begin{aligned}\frac{dx}{dt} &= ax - bxy \\ \frac{dy}{dt} &= dxy - cy,\end{aligned}$$

where a , b , c , and d are positive constants. x represents the size of the prey population at time t , whereas y represents the size of the predator population at time t . We use **Solve** to calculate the rest points. In this case, there is one boundary rest point, $E_0 = (0, 0)$, and one interior rest point, $E_1 = (c/d, a/b)$.

rps = Solve[[ax - bxy == 0, dxy - cy == 0], {x, y}]

{ {x → 0, y → 0}, {x → $\frac{c}{d}$, y → $\frac{a}{b}$ } }

The Jacobian is then found using **D**.

jac = {D[ax - bxy, x], D[ax - bxy, y], D[dxy - cy, x], D[dxy - cy, y]};
MatrixForm[jac]

$$\begin{pmatrix} a - by & -bx \\ dy & -c + dx \end{pmatrix}$$

E_0 is unstable because one eigenvalue of $\mathbf{J}(E_0)$ is positive. For the linearized system, E_1 is a center because the eigenvalues of $\mathbf{J}(E_1)$ are complex conjugates.

Eigenvalues[jac/.rps[[2]]]

{ $-i\sqrt{a}\sqrt{c}$, $i\sqrt{a}\sqrt{c}$ }

In fact, E_1 is a center for the nonlinear system as illustrated in Figure 6.41, where we have used $a = 1$, $b = 2$, $c = 2$, and $d = 1$. Notice that there are multiple limit cycles around $E_1 = (1/2, 1/2)$.

Needs["VectorFieldPlots"];

pvf = VectorFieldPlot[[x - 2xy, 2xy - y], {x, 0, 2},

{y, 0, 2}, PlotPoints → 30];

numgraph[init_, opts___] := Module[{numsol},

numsol = NDSolve[[x'[t] == x[t] - 2x[t]y[t], y'[t] == 2x[t]y[t] - y[t],

x[0] == init[[1]], y[0] == init[[2]], {x[t], y[t]}, {t, 0, 50}];

ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 10},

PlotStyle → {{Thickness[.01]}},

opts]]

i1 = Table[numgraph[{i, i}], {i, 3/20, 1/2, 1/20}];

Show[i1, pvf, DisplayFunction -> \$DisplayFunction, PlotRange -> {{0, 2}, {0, 2}},

AspectRatio -> Automatic]

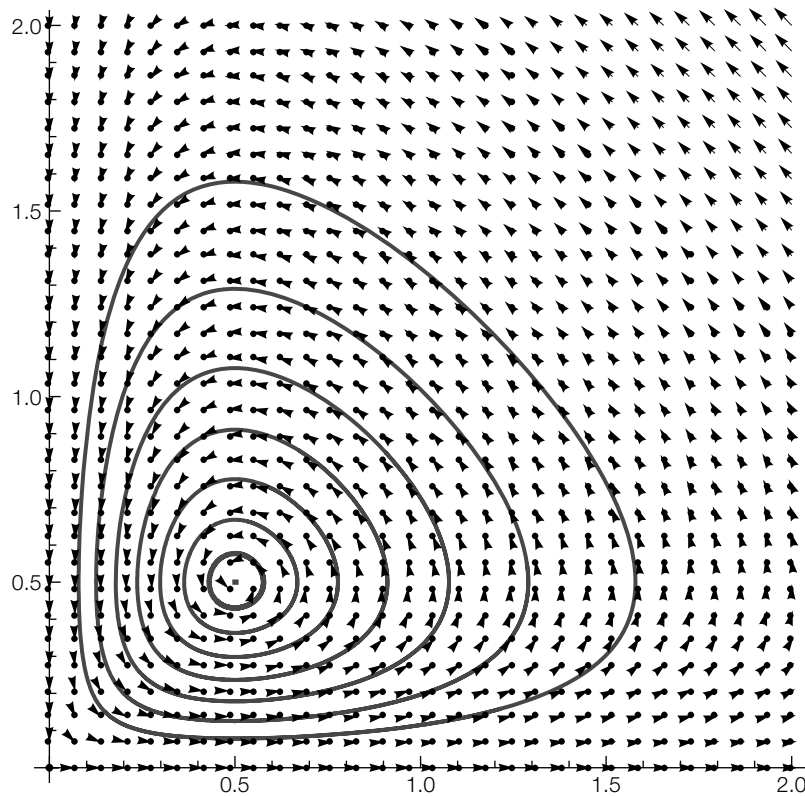
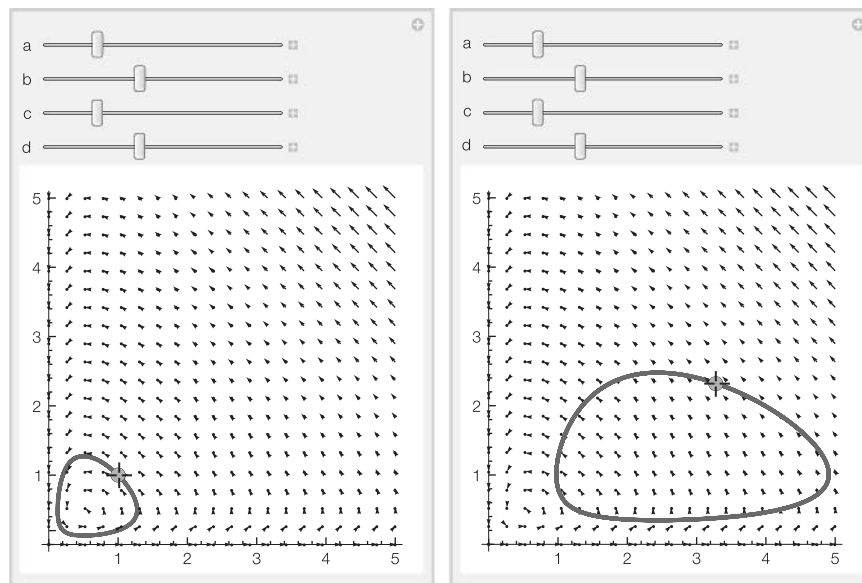


FIGURE 6.41

Multiple limit cycles about the interior rest point

As illustrated previously, if you want to play around with the system, use `Manipulate`. In this case, we allow a , b , c , and d to vary. The solution plotted is the one that passes through the locator point. See Figure 6.42.

```
Manipulate[
Needs["VectorFieldPlots"];
pvf = VectorFieldPlot[{ax - bxy, dxy - cy}, {x, 0, 5},
  {y, 0, 5}, PlotPoints -> 20];
numsol = NDSolve[{x'[t] == ax[t] - bx[t]y[t], y'[t] == dx[t]y[t] - cy[t],
  x[0] == init[[1]], y[0] == init[[2]]}, {x[t], y[t]}, {t, 0, 25}];
p1 = ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 25},
  PlotStyle -> {{Thickness[.01]}}];
Show[p1, pvf, PlotRange -> {{0, 5}, {0, 5}}, AspectRatio -> 1,
  AxesOrigin -> {0, 0},
  {{a, 1}, 0, 5}, {{b, 2}, 0, 5}, {{c, 1}, 0, 5}, {{d, 2}, 0, 5},
  {{init, {1, 1}}, Locator}]
```

**FIGURE 6.42**

Multiple limit cycles about the interior rest point

In this model, a stable interior rest state is not possible.

The complexity of the behavior of solutions to the system increases based on the assumptions made. Typical assumptions include adding satiation terms for the predator (y) and/or limiting the growth of the prey (x). The **standard predator–prey equations of Kolmogorov type**,

$$\begin{aligned} x' &= \alpha x \left(1 - \frac{1}{K}x \right) - \frac{mxy}{a+x} \\ y' &= y \left(\frac{mx}{a+x} - s \right), \end{aligned} \quad (6.40)$$

incorporate both of these assumptions.

We use **Solve** to find the three rest points of system (6.40). Let $E_0 = (0, 0)$ and $E_1 = (k, 0)$ denote the two boundary rest points, and let E_2 represent the interior rest point.

```
rps = Solve[{αx(1 - 1/kx) - mxy/(a + x) == 0, y(mx/(a + x) - s) == 0}, {x, y}]
{ {x -> 0, y -> 0}, {y -> 0, x -> k}, {y -> (akm - a^2s - aks)α / (k(m - s)^2), x -> -as / (-m + s)} }
```

The Jacobian, **J**, is calculated next in **s1**.

```
s1 = {{D[αx(1 - 1/kx) - mxy/(a + x), x], D[αx(1 - 1/kx) - mxy/(a + x), y]},
{D[y(mx/(a + x) - s), x], D[y(mx/(a + x) - s), y]}};
MatrixForm[s1]
```

$$\begin{pmatrix} \frac{mxy}{(a+x)^2} - \frac{my}{a+x} - \frac{x\alpha}{k} + \left(1 - \frac{x}{k}\right)\alpha & -\frac{mx}{a+x} \\ \left(-\frac{mx}{(a+x)^2} + \frac{m}{a+x}\right)y & -s + \frac{mx}{a+x} \end{pmatrix}$$

Because $\mathbf{J}(E_0)$ has one positive eigenvalue, E_0 is unstable.

```
e0 = s1/.rps[[1]];  
MatrixForm[e0]  
eigs0 = Eigenvalues[e0]
```

$$\begin{pmatrix} \alpha & 0 \\ 0 & -s \end{pmatrix}$$

$$\{-s, \alpha\}$$

The stability of E_1 is determined by the sign of $m - s - am/(a + k)$.

```
e1 = s1/.rps[[2]];  
MatrixForm[e1]  
eigs1 = Eigenvalues[e1]
```

$$\begin{pmatrix} -\alpha & -\frac{km}{a+k} \\ 0 & \frac{km}{a+k} - s \end{pmatrix}$$

$$\left\{\frac{km}{a+k} - s, -\alpha\right\}$$

The eigenvalues of $\mathbf{J}(E_2)$ are quite complicated.

```
e2 = s1/.rps[[3]];  
MatrixForm[e2]  
eigs2 = Eigenvalues[e2]
```

$$\begin{pmatrix} \frac{as\alpha}{k(-m+s)} - \frac{ams(akm - a^2s - aks)\alpha}{k(m-s)^2(-m+s)\left(a - \frac{as}{-m+s}\right)^2} - \frac{m(akm - a^2s - aks)\alpha}{k(m-s)^2\left(a - \frac{as}{-m+s}\right)} + \left(1 + \frac{as}{k(-m+s)}\right)\alpha & \frac{ams}{(-m+s)\left(a - \frac{as}{-m+s}\right)} \\ \frac{(akm - a^2s - aks)\left(\frac{ams}{(-m+s)\left(a - \frac{as}{-m+s}\right)^2} + \frac{m}{a - \frac{as}{-m+s}}\right)\alpha}{k(m-s)^2} & -s - \frac{ams}{(-m+s)\left(a - \frac{as}{-m+s}\right)} \end{pmatrix}$$

$$\left\{ \frac{1}{2(-km^2 + kms)} \left(ams\alpha - kms\alpha + as^2\alpha + ks^2\alpha - \sqrt{\left((-ams\alpha + kms\alpha - as^2\alpha - ks^2\alpha)^2 - 4(-km^2 + kms)(-km^2s\alpha + ams^2\alpha + 2kms^2\alpha - as^3\alpha - ks^3\alpha) \right)} \right), \frac{1}{2(-km^2 + kms)} \right. \\ \left. \left(ams\alpha - kms\alpha + as^2\alpha + ks^2\alpha + \sqrt{\left((-ams\alpha + kms\alpha - as^2\alpha - ks^2\alpha)^2 - 4(-km^2 + kms)(-km^2s\alpha + ams^2\alpha + 2kms^2\alpha - as^3\alpha - ks^3\alpha) \right)} \right) \right\}$$

Instead of using the eigenvalues, we compute the characteristic polynomial of $\mathbf{J}(E_2)$, $p(\lambda) = c_2\lambda^2 + c_1\lambda + c_0$, and examine the coefficients. Notice that c_2 is always positive.

```
cpe2 = CharacteristicPolynomial[e2, λ]//Simplify  

$$\frac{as\alpha(m(-s + \lambda) + s(s + \lambda)) + k(m - s)(-s\alpha(s + \lambda) + m(s\alpha + \lambda^2))}{km(m - s)}$$

```

c0 = cpe2/.λ->0//Simplify

$$\frac{s(k(m-s)-as)\alpha}{km}$$

c1 = Coefficient[cpe2, λ]//Simplify

$$\frac{s(k(-m+s)+a(m+s))\alpha}{km(m-s)}$$

c2 = Coefficient[cpe2, λ^2]//Simplify

1

On the other hand, c_0 and $m - s - am/(a + k)$ have the same sign because

c0/eigs1[[1]]//Simplify

$$\frac{(a+k)s\alpha}{km}$$

is always positive. In particular, if $m - s - am/(a + k) < 0$, E_1 is stable. Because c_0 is negative, by Descartes' rule of signs, it follows that $p(\lambda)$ will have one positive root and hence E_2 will be unstable.

On the other hand, if $m - s - am/(a + k) > 0$ so that E_1 is unstable, E_2 may be either stable or unstable. To illustrate these two possibilities, let $\alpha = K = m = 1$ and $a = 1/10$. We recalculate.

α = 1; k = 1; m = 1; a = 1/10;

rps = Solve[{αx(1 - 1/kx) - mxy/(a + x) == 0, y(mx/(a + x) - s) == 0}, {x, y}]

$$\left\{ \left\{ x \rightarrow 0, y \rightarrow 0 \right\}, \left\{ y \rightarrow 0, x \rightarrow 1 \right\}, \left\{ y \rightarrow \frac{10-11s}{100(-1+s)^2}, x \rightarrow -\frac{s}{10(-1+s)} \right\} \right\}$$

**s1 = {{D[αx(1 - 1/kx) - mxy/(a + x), x], D[αx(1 - 1/kx) - mxy/(a + x), y]},
{D[y(mx/(a + x) - s), x], D[y(mx/(a + x) - s), y]}};**

MatrixForm[s1]

$$\begin{pmatrix} 1 - 2x + \frac{xy}{\left(\frac{1}{10} + x\right)^2} - \frac{y}{\frac{1}{10} + x} & -\frac{x}{\frac{1}{10} + x} \\ \left(-\frac{x}{\left(\frac{1}{10} + x\right)^2} + \frac{1}{\frac{1}{10} + x}\right)y & -s + \frac{x}{\frac{1}{10} + x} \end{pmatrix}$$

e2 = s1/.rps[[3]];

cpe2 = CharacteristicPolynomial[e2, λ]//Simplify

$$\frac{-11s^3 + s^2(21 - 11\lambda) - 10\lambda^2 + s(-10 + 9\lambda + 10\lambda^2)}{10(-1 + s)}$$

c0 = cpe2/.λ->0//Simplify

$$s - \frac{11s^2}{10}$$

c1 = Coefficient[cpe2, λ]//Simplify

$$\frac{(9 - 11s)s}{10(-1 + s)}$$

c2 = Coefficient[cpe2, λ^2]//Simplify

1

Using Reduce, we see that

1. c_0 , c_1 , and c_2 are positive if $9/11 < s < 10/11$, and
2. c_0 and c_2 are positive and c_1 is negative if $0 < s < 9/11$.

Reduce[$c_0 > 0 \&\& c_1 > 0, s]$

$$\frac{9}{11} < s < \frac{10}{11}$$

Reduce[$c_0 > 0 \&\& c_1 < 0, s]$

$$0 < s < \frac{9}{11}$$

In the first situation, E_2 is stable; in the second, E_2 is unstable.

Using $s = 19/22$, we graph the direction field associated with the system as well as various solutions in Figure 6.43(a). In the plot, notice that all nontrivial solutions approach $E_2 \approx (0.63, 0.27)$; E_2 is stable—a situation that cannot occur with the standard predator–prey equations.

rps/.s->19/22//N

{ {x → 0., y → 0.}, {y → 0., x → 1.}, {y → 0.268889, x → 0.633333} }

Needs["VectorFieldPlots"]

Clear[pvf, numgraph, i1, i2]

pvf = VectorFieldPlot [$\left\{ \alpha x \left(1 - \frac{x}{k} \right) - \frac{mxy}{a+x}, y \left(\frac{mx}{a+x} - \frac{19}{22} \right) \right\}$, {x, 0, 1},
{y, 0, 1}, PlotPoints → 30];

numgraph[init_, s_, opts_] := Module[{numsol},

numsol = NDSolve[{x'[t] == $\alpha x[t] \left(1 - \frac{1}{kx[t]} \right) - mx[t]y[t]/(a + x[t])$,
y'[t] == $y[t] \left(mx[t]/(a + x[t]) - s \right)$, x[0] == init[[1]], y[0] == init[[2]]},
{x[t], y[t]}, {t, 0, 50}];

ParametricPlot[Evaluate[{x[t], y[t]}.numsol], {t, 0, 50},
PlotStyle → Thickness[.01], opts]]

i1 = Table[numgraph[{i, 1}, 19/22], {i, 0, 1, 1/10}];

i2 = Table[numgraph[{i, 1}, 19/22], {i, 0, 1, 1/10}];

Show[i1, i2, pvf, PlotRange → {{0, 1}, {0, 1}}, AspectRatio → Automatic]

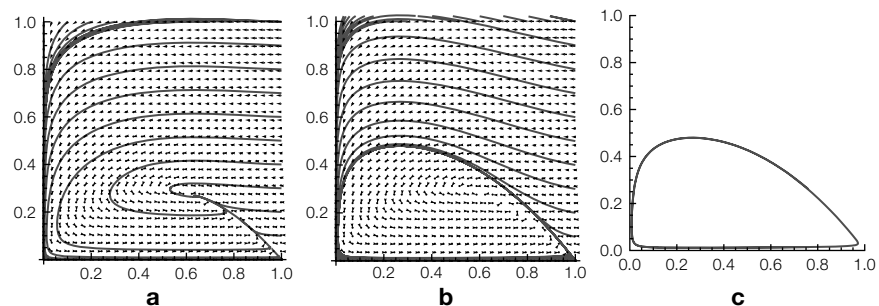


FIGURE 6.43

(a) $s = 19/22$. (b) $s = 8/11$. (c) A better view of the limit cycle without the direction field

On the other hand, using $s = 8/11$ (so that E_2 is unstable) in Figure 6.43(b), we see that all nontrivial solutions appear to approach a limit cycle.

```
rps/.s->8/11//N
{{x -> 0., y -> 0.}, {y -> 0., x -> 1.}, {y -> 0.268889, x -> 0.266667}}

i1 = Table[numgraph[{1, i}, 8/11], {i, 0, 1, 1/10}];
i2 = Table[numgraph[{i, 1}, 8/11], {i, 0, 1, 1/10}];
p1 = Show[i1, i2, pvf, PlotRange->{{0, 1}, {0, 1}}, AspectRatio->Automatic]
```

The limit cycle is shown more clearly in Figure 6.43(c).

```
numgraph[{.759, .262}, 8/11, PlotRange->{{0, 1}, {0, 1}},
  AspectRatio->Automatic]
```

As we have seen in similar situations, these commands can be collected into a single Manipulate command to investigate the situation. See Figure 6.44.

```
Clear[pvf, numgraph, i1, i2,  $\alpha$ , k, m]
Manipulate[
Needs["VectorFieldPlots"];
pvf = VectorFieldPlot[{ $\alpha x (1 - \frac{x}{k}) - \frac{mxy}{a+x}$ ,  $y (\frac{mx}{a+x} - s)$ }, {x, 0, 1},
{y, 0, 1}, PlotPoints -> 20];
numgraph[init_, s_] := Module[{numsol},
```

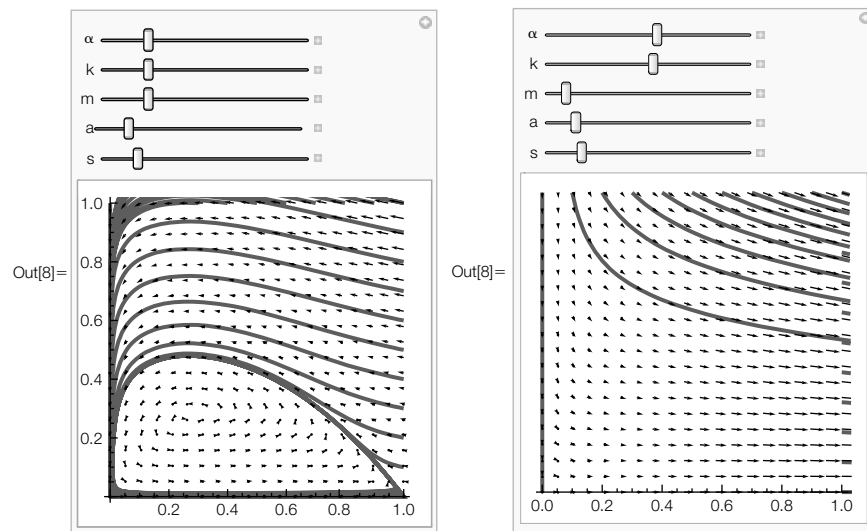


FIGURE 6.44

Using Manipulate to investigate the standard predator-prey equations of Komogorov type

```

numsol = NDSolve[{x'[t]==αx[t](1-1/kx[t])-mx[t]y[t]/(a+x[t]),
  y'[t]==y[t](mx[t]/(a+x[t])-s), x[0]==init[[1]],
  y[0]==init[[2]]},
  {x[t], y[t]}, {t, 0, 50}];
ParametricPlot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 50},
  PlotPoints → 200,
  PlotStyle → Thickness[.01]];
i1 = Table[numgraph[{1, i}, s], {i, 0, 1, 1/10}];
i2 = Table[numgraph[{i, 1}, s], {i, 0, 1, 1/10}];
Show[i1, i2, pvf, PlotRange->{{0, 1}, {0, 1}}, AspectRatio->Automatic],
  {{α, 1}, 0, 5}, {{k, 1}, 0, 5}, {{m, 1}, 0, 5}, {{a, 1/10}, 0, 1},
  {{s, 8/11}, 0, 5}]

```

Example 6.4.8 (Van der Pol's equation). In Example 6.4.5, we saw that **Van der Pol's equation**,

Also see Example 6.4.5.

$x'' + \mu(x^2 - 1)x' + x = 0$, is equivalent to the system $\begin{cases} x' = y \\ y' = \mu(1 - x^2)y - x \end{cases}$.

Classify the equilibrium points, use NDSolve to approximate the solutions to this nonlinear system, and plot the phase plane.

Solution We find the equilibrium points by solving $\begin{cases} y = 0 \\ \mu(1 - x^2)y - x = 0 \end{cases}$. From the first equation, we see that $y = 0$. Then, substitution of $y = 0$ into the second equation yields $x = 0$. Therefore, the only equilibrium point is $(0, 0)$. The Jacobian matrix for this system is

$$J(x, y) = \begin{pmatrix} 0 & 1 \\ -1 - 2\mu xy & -\mu(x^2 - 1) \end{pmatrix}.$$

The eigenvalues of $J(0, 0)$ are $\lambda_{1,2} = \frac{1}{2}(\mu \pm \sqrt{\mu^2 - 4})$.

```

Clear[f, g]
f[x_, y_] = y;
g[x_, y_] = -x - μ (x^2 - 1) y;
jac =  $\begin{pmatrix} D[f[x, y], x] & D[f[x, y], y] \\ D[g[x, y], x] & D[g[x, y], y] \end{pmatrix}$ ;
jac/.{x->0, y->0}/Eigenvalues
 $\left\{ \frac{1}{2}(\mu - \sqrt{-4 + \mu^2}), \frac{1}{2}(\mu + \sqrt{-4 + \mu^2}) \right\}$ 

```

Notice that if $\mu > 2$, then both eigenvalues are positive and real. Hence, we classify $(0, 0)$ as an **unstable node**. On the other hand, if $0 < \mu < 2$, then the eigenvalues are a complex conjugate pair with a positive real part. Hence, $(0, 0)$ is an **unstable spiral**. (We omit the case $\mu = 2$ because the eigenvalues are repeated.)

We now show several curves in the phase plane that begin at various points for various values of μ . First, we define the function `sol`, which given μ , x_0 , and y_0 , generates a numerical solution to the initial-value problem

$$\begin{cases} x' = y \\ y' = \mu(1 - x^2)y - x \\ x(0) = x_0, y(0) = y_0, \end{cases}$$

and then parametrically graphs the result for $0 \leq t \leq 20$.

```
Clear[sol]
sol[μ_, {x0_, y0_}, opts_] :=
Module[{eqone, eqtwo, solt}, eqone = x'[t]==y[t];
  eqtwo = y'[t]==μ (1 - x[t]^2) y[t] - x[t];
  solt = NDSolve[{eqone, eqtwo, x[0]==x0, y[0]==y0}, {x[t], y[t]},
    {t, 0, 20}];
  ParametricPlot[{x[t], y[t]}/.solt, {t, 0, 20}, opts]]
```

We then use `Table` and `Union` to generate a list of ordered pairs `initconds` that will correspond to the initial conditions in the initial-value problem.

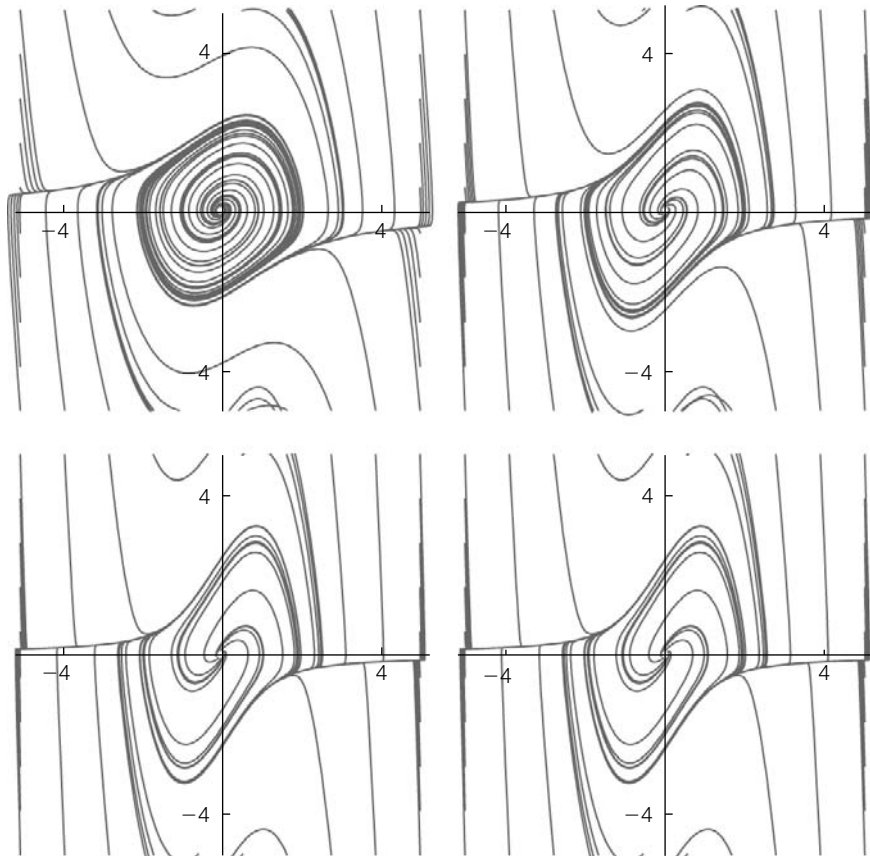
```
initconds1 = Table[{0.1Cos[t], 0.1Sin[t]}, {t, 0, 2π, 2π/9}];
initconds2 = Table[{-5, i}, {i, -5, 5, 10/9}];
initconds3 = Table[{5, i}, {i, -5, 5, 10/9}];
initconds4 = Table[{i, 5}, {i, -5, 5, 10/9}];
initconds5 = Table[{i, -5}, {i, -5, 5, 10/9}];
initconds = initconds1 ∪ initconds2 ∪ initconds3 ∪ initconds4 ∪ initconds5;
```

Last, we use `Map` to apply `sol` to the list of ordered pairs in `initconds` for $\mu = 1/2$.

```
somegraphs1 = Map[sol[1/2, #, DisplayFunction->Identity]&, initconds];
phase1 = Show[somegraphs1, PlotRange → {{-5, 5}, {-5, 5}},
  AspectRatio → 1, Ticks → {{-4, 4}, {-4, 4}}]
```

Similarly, we use `Map` to apply `sol` to the list of ordered pairs in `initconds` for $\mu = 1$, $3/2$, and 3 .

```
somegraphs2 = Map[sol[1, #, DisplayFunction->Identity]&, initconds];
phase2 = Show[somegraphs2, PlotRange → {{-5, 5}, {-5, 5}},
  AspectRatio → 1, Ticks → {{-4, 4}, {-4, 4}}]
somegraphs3 = Map[sol[3/2, #, DisplayFunction->Identity]&, initconds];
phase3 = Show[somegraphs3, PlotRange → {{-5, 5}, {-5, 5}},
  AspectRatio → 1, Ticks → {{-4, 4}, {-4, 4}}]
somegraphs4 = Map[sol[3, #, DisplayFunction->Identity]&, initconds];
phase4 = Show[somegraphs3, PlotRange → {{-5, 5}, {-5, 5}},
  AspectRatio → 1, Ticks → {{-4, 4}, {-4, 4}}]
```

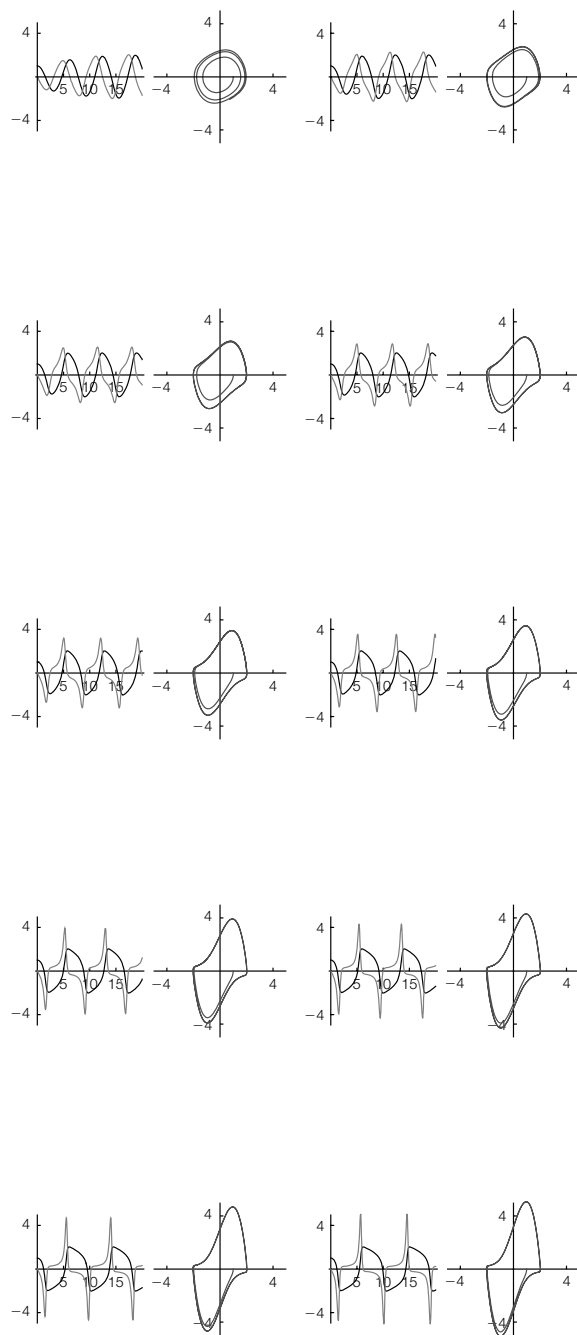
**FIGURE 6.45**

Solutions to the Van der Pol equation for various values of μ

All four graphs are shown together in Figure 6.45. In each figure, we see that all of the curves approach a curve called a *limit cycle*. Physically, the fact that the system has a limit cycle indicates that for all oscillations, the motion eventually becomes periodic, which is represented by a closed curve in the phase plane.

Show[GraphicsGrid[{{phase1, phase2}, {phase3, phase4}}]]

On the other hand, in Figure 6.46 we graph the solutions that satisfy the initial conditions $x(0) = 1$ and $y(0) = 0$ parametrically and individually for various values of μ . Notice that for small values of μ the system more closely approximates that of the harmonic oscillator because the damping coefficient is small. The curves are more circular than those for larger values of μ .

**FIGURE 6.46**

The solutions to the Van der Pol equation satisfying $x(0) = 1$ and $y(0) = 0$ individually (x in black and y in gray) for various values of μ

```

Clear[x, y, t, s]
graph[μ_]:=Module[{numsol, pp, pxy},
  numsol = NDSolve[{x'[t]==y[t], y'[t]==μ(1-x[t]^2)y[t]-x[t],
    x[0]==1, y[0]==0}, {x[t], y[t]}, {t, 0, 20}];
  pp = ParametricPlot[{x[t], y[t]}/.numsol, {t, 0, 20},
    PlotRange -> {{-5, 5}, {-5, 5}}, AspectRatio -> 1,
    Ticks -> {{-4, 4}, {-4, 4}}, DisplayFunction -> Identity];
  pxy = Plot[Evaluate[{x[t], y[t]}/.numsol], {t, 0, 20},
    PlotStyle -> {GrayLevel[0], GrayLevel["0.5"]}, PlotRange -> {-5, 5},
    AspectRatio -> 1, Ticks -> {{5, 10, 15}, {-4, 4}},
    DisplayFunction -> Identity];
  GraphicsRow[{pxy, pp}]]
graphs = Table[graph[i], {i, 0.25, 3, 2.75/9}];
toshow = Partition[graphs, 2];
Show[GraphicsGrid[toshow]]

```

An alternative to comparing the graphics together is to use `Manipulate` to create an animation of how the μ values affect the solutions of the equation. See Figure 6.47.

```

Manipulate[
sol[μ_, {x0_, y0_}, opts_]:=
  Module[{eqone, eqtwo, solt}, eqone = x'[t]==y[t];
  eqtwo = y'[t]==μ(1-x[t]^2)y[t]-x[t];
  solt = NDSolve[{eqone, eqtwo, x[0]==x0, y[0]==y0}, {x[t], y[t]}, {t, 0, 20}];

```

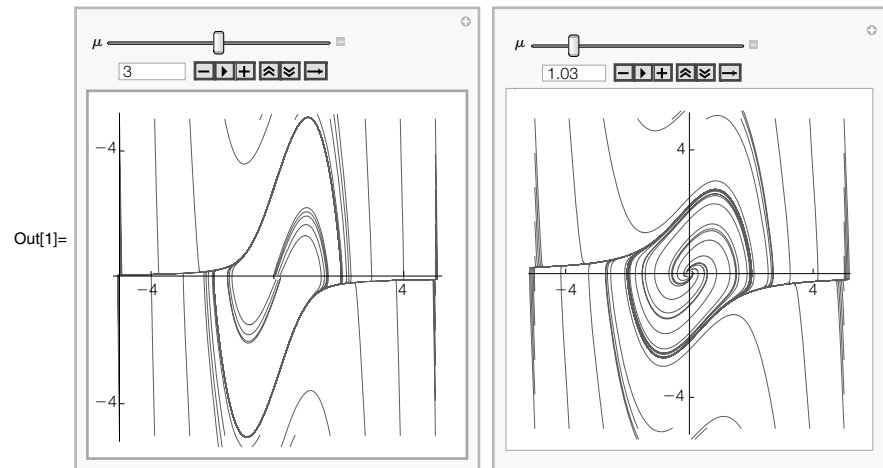


FIGURE 6.47

Varying μ in the Van der Pol equation with `Manipulate`

```

ParametricPlot[{x[t], y[t]}/.solt, {t, 0, 20}, PlotPoints -> 200];
initconds1 = Table[{0.1Cos[t], 0.1Sin[t]}, {t, 0, 2π, 2π/9}];
initconds2 = Table[{-5, i}, {i, -5, 5, 10/9}];
initconds3 = Table[{5, i}, {i, -5, 5, 10/9}];
initconds4 = Table[{i, 5}, {i, -5, 5, 10/9}];
initconds5 = Table[{i, -5}, {i, -5, 5, 10/9}];
initconds = initconds1 ∪ initconds2 ∪ initconds3 ∪ initconds4 ∪ initconds5;
somegraphs1 = Map[sol[μ, #, DisplayFunction->Identity]&, initconds];
phase1 = Show[somegraphs1, PlotRange -> {{-5, 5}, {-5, 5}},
  AspectRatio -> 1, Ticks -> {{-4, 4}, {-4, 4}}, {{μ, 3}, 0, 6}]

```

Although linearization can help you determine local behavior near rest points, the long-term behavior of solutions to nonlinear systems can be quite complicated, even for deceptively simple looking systems.

Example 6.4.9 (Lorenz Equations). The **Lorenz equations** are

$$\begin{cases} dx/dt = a(y - x) \\ dy/dt = bx - y - xz \\ dz/dt = xy - cz \end{cases}.$$

Graph the solutions to the Lorenz equations if $a = 7$, $b = 27.2$, and $c = 3$ if the initial conditions are $x(0) = 3$, $y(0) = 4$, and $z(0) = 2$.

Solution So that you can experiment with different parameters and initial conditions, we use **Manipulate** to solve the Lorenz system using initial conditions $x(0) = x_0$, $y(0) = y_0$, and $z(0) = z_0$ for $950 \leq t \leq 1000$; generate parametric plots of x versus y , y versus z , x versus z , and x versus y versus z ; and display the four resulting plots as a graphics array.

See texts such as Jordan and Smith's *Nonlinear Ordinary Differential Equations* [23] for discussions of ways to analyze systems such as the Rössler attractor and the Lorenz equations.

Because the behavior of solutions can be quite intricate, we include the option **MaxSteps->Infinity** in the **NDSolve** command to help Mathematica capture the oscillatory behavior in the long-term solution. See Figure 6.48.

On the other hand, if you define **lorenzsol** separately,

```

Clear[x, y, z, lorenzsol]
lorenzsol[a_, b_, c_][{x0_, y0_, z0_}, ts_:{t, 0, 1000},
  opts_]:=Module[{numsol},
  numsol =
  NDSolve[{x'[t]==-ax[t]+ay[t],
    y'[t]==bx[t]-y[t]-x[t]z[t],
    z'[t]==x[t]y[t]-cz[t], x[0]==x0,
    y[0]==y0, z[0]==z0}, {x[t], y[t], z[t]},
    ts, MaxSteps -> Infinity]

```

]

The Lorenz Equations

```
Manipulate[
Clear[x, y, z, lorenzsol];
lorenzsol[a_, b_, c_] := Module[{x0_, y0_, z0_, ts_ := {t, 0, 1000},
opts_ := Module[{numsol},
numsol =
NDSolve[{x'[t] == -a x[t] + a y[t],
y'[t] == b x[t] - y[t] - x[t] z[t],
z'[t] == x[t] y[t] - c z[t], x[0] == x0,
y[0] == y0, z[0] == z0}, {x[t], y[t], z[t]},
ts, MaxSteps -> Infinity];
n1 = lorenzsol[a, b, c][{x0, y0, z0}];
pla = ParametricPlot[Evaluate[{x[t], y[t]} /. n1],
{t, 900, 950}, PlotPoints -> 1000, AspectRatio -> 1,
AxesLabel -> {"x", "y"}];
plb = ParametricPlot[Evaluate[{x[t], z[t]} /. n1],
{t, 900, 950}, PlotPoints -> 1000, AspectRatio -> 1,
AxesLabel -> {"x", "z"}];
plc = ParametricPlot[Evaluate[{y[t], z[t]} /. n1],
{t, 900, 950}, PlotPoints -> 1000, AspectRatio -> 1,
AxesLabel -> {"y", "z"}];
pld = ParametricPlot3D[
Evaluate[{x[t], y[t], z[t]} /. n1], {t, 950, 1000},
PlotPoints -> 3000, BoxRatios -> {1, 1, 1},
AxesLabel -> {"x", "y", "z"}];
Show[GraphicsGrid[{{pla, plb}, {plc, pld}}], {{a, 7}, {b, 9},
{{b, 27.2}, {25, 30}}, {{c, 3}, {1, 5}}, {{x0, 3}, {0, 5}}, {{y0, 4}, {0, 5}},
{{z0, 2}, {0, 5}}]
```

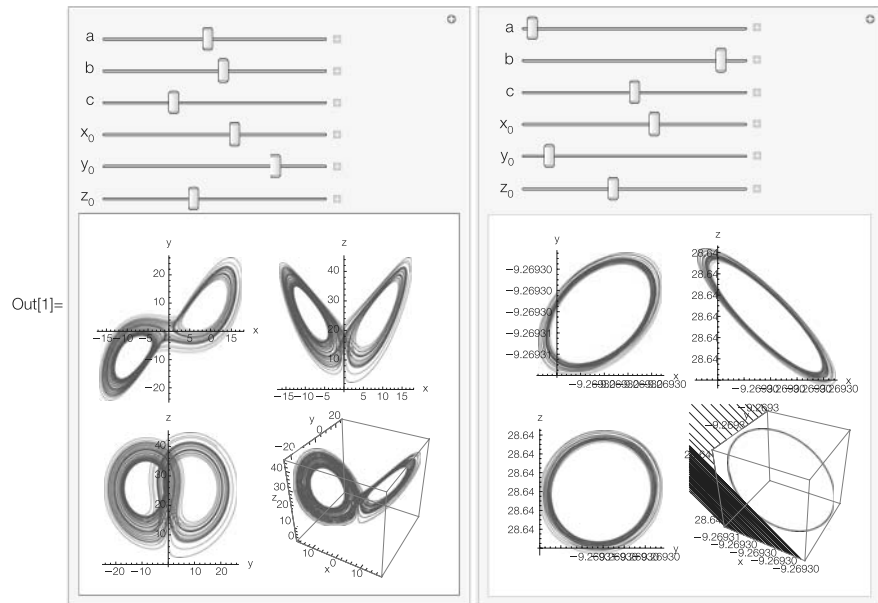
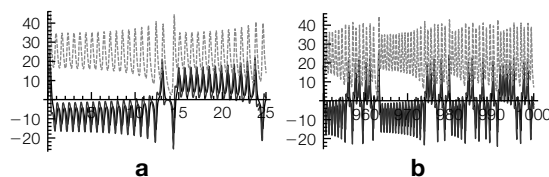


FIGURE 6.48

Comparing a chaotic solution to a non-chaotic solution with **Manipulate**

we can then use `lorenzplot` to generate a numerical solution for our parameter values and initial conditions.

```
n2 = lorenzsol[7, 27.2, 3][{3, 4, 2}];
```

**FIGURE 6.49**

(a) Plots of $x(t)$ (in black), $y(t)$ (in gray), and $z(t)$ (dashed) for $0 \leq t \leq 25$. (b) Plots of $x(t)$ (in black), $y(t)$ (in gray), and $z(t)$ (dashed) for $950 \leq t \leq 1000$

We generate a short-term plot of the solution in Figure 6.49(a) and a long-term plot in Figure 6.49(b).

```
pp1 = Plot[Evaluate[{x[t], y[t], z[t]}/.n2], {t, 0, 25},
  PlotStyle → {GrayLevel[0], GrayLevel[.3], Dashing[{0.01]}],
  PlotPoints → 1000];

pp2 = Plot[Evaluate[{x[t], y[t], z[t]}/.n2], {t, 950, 1000},
  PlotStyle → {GrayLevel[0], GrayLevel[.3], Dashing[{0.01]}],
  PlotPoints → 1000];

Show[GraphicsRow[{pp1, pp2}]]
```

6.5 SOME PARTIAL DIFFERENTIAL EQUATIONS

6.5.1 The One-Dimensional Wave Equation

Suppose that we pluck a string (such as a guitar or violin string) of length p and constant mass density that is fixed at each end. A question that we might ask is: What is the position of the string at a particular instance of time? We answer this question by modeling the physical situation with a partial differential equation, namely the wave equation in one spatial variable:

$$c^2 \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2} \quad \text{or} \quad c^2 u_{xx} = u_{tt}. \quad (6.41)$$

In equation (6.41), $c^2 = T/\rho$, where T is the tension of the string and ρ is the constant mass of the string per unit length. The solution $u(x, t)$ represents the displacement of the string from the x -axis at time t . To determine u , we must describe the boundary and initial conditions that model the physical situation. At the ends of the string, the displacement from the x -axis is fixed at zero, so we use the homogeneous boundary conditions $u(0, t) = u(p, t) = 0$

for $t > 0$. The motion of the string also depends on the displacement and the velocity at each point of the string at $t = 0$. If the initial displacement is given by $f(x)$ and the initial velocity by $g(x)$, we have the initial conditions $u(x, 0) = f(x)$ and $u_t(x, 0) = g(x)$ for $0 \leq x \leq p$. Therefore, we determine the displacement of the string with the initial boundary value problem

$$\begin{cases} c^2 \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2}, & 0 < x < p, t > 0 \\ u(0, t) = u(p, t) = 0, & t > 0 \\ u(x, 0) = f(x), u_t(x, 0) = g(x), & 0 < x < p. \end{cases} \quad (6.42)$$

This problem is solved through separation of variables by assuming that $u(x, t) = X(x)T(t)$. Substitution into equation (6.41) yields

λ is a constant.

$$c^2 X'' T = X T'' \quad \text{or} \quad \frac{X''}{X} = \frac{T''}{c^2 T} = -\lambda,$$

so we obtain the two second-order ordinary differential equations $X'' + \lambda X = 0$ and $T'' + c^2 \lambda T = 0$. At this point, we solve the equation that involves the homogeneous boundary conditions. The boundary conditions in terms of $u(x, t) = X(x)T(t)$ are $u(0, t) = X(0)T(t) = 0$ and $u(p, t) = X(p)T(t) = 0$, so we have $X(0) = 0$ and $X(p) = 0$. Therefore, we determine $X(x)$ by solving the *eigenvalue problem*

$$\begin{cases} X'' + \lambda X = 0, & 0 < x < p \\ X(0) = X(p) = 0. \end{cases}$$

The eigenvalues of this problem are $\lambda_n = (n\pi/p)^2$, $n = 1, 3, \dots$ with corresponding eigenfunctions $X_n(x) = \sin(n\pi x/p)$, $n = 1, 3, \dots$. Next, we solve the equation $T'' + c^2 \lambda_n T = 0$. A general solution is

$$T_n(t) = a_n \cos\left(c\sqrt{\lambda_n}t\right) + b_n \sin\left(c\sqrt{\lambda_n}t\right) = a_n \cos \frac{cn\pi t}{p} + b_n \sin \frac{cn\pi t}{p},$$

where the coefficients a_n and b_n must be determined. Putting this information together, we obtain

$$u_n(x, t) = \left(a_n \cos \frac{cn\pi t}{p} + b_n \sin \frac{cn\pi t}{p}\right) \sin \frac{n\pi x}{p},$$

so by the principle of superposition, we have

$$u(x, t) = \sum_{n=1}^{\infty} \left(a_n \cos \frac{cn\pi t}{p} + b_n \sin \frac{cn\pi t}{p}\right) \sin \frac{n\pi x}{p}.$$

Applying the initial displacement $u(x, 0) = f(x)$ yields

$$u(x, 0) = \sum_{n=1}^{\infty} a_n \sin \frac{n\pi x}{p} = f(x),$$

so a_n is the *Fourier sine series coefficient* for $f(x)$, which is given by

$$a_n = \frac{2}{p} \int_0^p f(x) \sin \frac{n\pi x}{p} dx, \quad n = 1, 2, \dots$$

In order to determine b_n , we must use the initial velocity. Therefore, we compute

$$\frac{\partial u}{\partial t}(x, t) = \sum_{n=1}^{\infty} \left(-a_n \frac{cn\pi}{p} \sin \frac{cn\pi t}{p} + b_n \frac{cn\pi}{p} \cos \frac{cn\pi t}{p} \right) \sin \frac{n\pi x}{p}.$$

Then,

$$\frac{\partial u}{\partial t}(x, 0) = \sum_{n=1}^{\infty} b_n \frac{cn\pi}{p} \sin \frac{n\pi x}{p} = g(x)$$

so $b_n \frac{cn\pi}{p}$ represents the Fourier sine series coefficient for $g(x)$, which means that

$$b_n = \frac{p}{cn\pi} \int_0^p g(x) \sin \frac{n\pi x}{p} dx, \quad n = 1, 2, \dots$$

Example 6.5.1 Solve $\begin{cases} u_{xx} = u_t, & 0 < x < 1, t > 0 \\ u(0, t) = u(1, t) = 0, & t > 0 \\ u(x, 0) = \sin \pi x, & u_t(x, 0) = 3x + 1, & 0 < x < 1. \end{cases}$

Solution The initial displacement and velocity functions are defined first.

$$f[x_] = \text{Sin}[\pi x];$$

$$g[x_] = 3x + 1;$$

Next, the functions to determine the coefficients a_n and b_n in the series approximation of the solution $u(x, t)$ are defined. Here, $p = c = 1$.

$$a_1 = 2 \int_0^1 f[x] \text{Sin}[\pi x] dx$$

1

$$a_{n_} = 2 \int_0^1 f[x] \text{Sin}[n\pi x] dx$$

$$\frac{2 \text{Sin}[n\pi]}{\pi - n^2 \pi}$$

$$b_{n_} = \frac{2 \int_0^1 g[x] \text{Sin}[n\pi x] dx}{n\pi} // \text{Simplify}$$

$$\frac{2n\pi - 8n\pi \text{Cos}[n\pi] + 6 \text{Sin}[n\pi]}{n^3 \pi^3}$$

Because n represents an integer, these results indicate that $a_n = 0$ for all $n \geq 2$, which we confirm with **Simplify** together with the **Assumptions** by instructing Mathematica to assume that n is an integer.

$$\text{Simplify} \left[\frac{2 \text{Sin}[n\pi]}{\pi - n^2 \pi}, \text{Assumptions} \rightarrow \text{Element}[n, \text{Integers}] \right]$$

0

Simplify $\left[\frac{2n\pi - 8n\pi \cos[n\pi] + 6\sin[n\pi]}{n^3\pi^3}, \right.$
Assumptions $\rightarrow$ **Element**[n , Integers]

$$\frac{2 - 8(-1)^n}{n^2\pi^2}$$

We use **Table** to calculate the first 10 values of b_n .

Table[{ n , b_n , b_n/N }, { n , 1, 10}] // **TableForm**

1	$\frac{10}{\pi^2}$	1.01321
2	$-\frac{3}{2\pi^2}$	-0.151982
3	$\frac{10}{9\pi^2}$	0.112579
4	$-\frac{3}{8\pi^2}$	-0.0379954
5	$\frac{2}{5\pi^2}$	0.0405285
6	$-\frac{1}{6\pi^2}$	-0.0168869
7	$\frac{10}{49\pi^2}$	0.0206778
8	$-\frac{3}{32\pi^2}$	-0.00949886
9	$\frac{10}{81\pi^2}$	0.0125088
10	$-\frac{3}{50\pi^2}$	-0.00607927

Notice that we define **uapprox**[n] so that Mathematica “remembers” the terms **uapprox** that are computed. That is, Mathematica does not need to recompute **uapprox**[$n-1$] to compute **uapprox**[n] provided that **uapprox**[$n-1$] has already been computed.

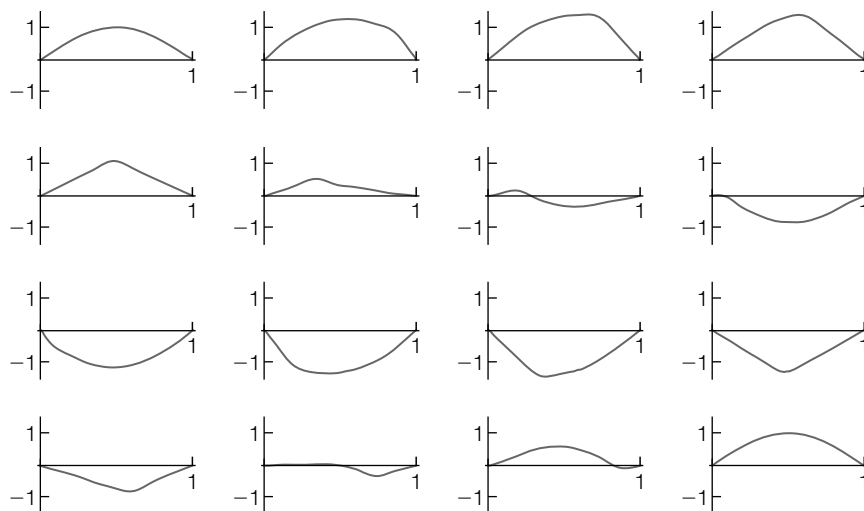
The function **u** defined next computes the n th term in the series expansion. Thus, **uapprox** determines the approximation of order k by summing the first k terms of the expansion, as illustrated with **approx**[10].

Clear[**u**, **uapprox**]
u[$n_$] = $b_n \sin[n\pi t] \sin[n\pi x]$;
uapprox[$k_$] := **uapprox**[k] = **uapprox**[$k-1$] + **u**[k];
uapprox[0] = **Cos**[πt] **Sin**[πx];
uapprox[10]

$$\begin{aligned} & \cos[\pi t] \sin[\pi x] + \frac{10 \sin[\pi t] \sin[\pi x]}{\pi^2} - \frac{3 \sin[2\pi t] \sin[2\pi x]}{2\pi^2} + \\ & \frac{10 \sin[3\pi t] \sin[3\pi x]}{9\pi^2} - \frac{3 \sin[4\pi t] \sin[4\pi x]}{8\pi^2} + \\ & \frac{2 \sin[5\pi t] \sin[5\pi x]}{5\pi^2} - \frac{\sin[6\pi t] \sin[6\pi x]}{6\pi^2} + \\ & \frac{10 \sin[7\pi t] \sin[7\pi x]}{49\pi^2} - \frac{3 \sin[8\pi t] \sin[8\pi x]}{32\pi^2} + \\ & \frac{10 \sin[9\pi t] \sin[9\pi x]}{81\pi^2} - \frac{3 \sin[10\pi t] \sin[10\pi x]}{50\pi^2} \end{aligned}$$

To illustrate the motion of the string, we graph **uapprox**[10], the 10th partial sum of the series, on the interval [0, 1] for 16 equally spaced values of t between 0 and 2 in Figure 6.50.

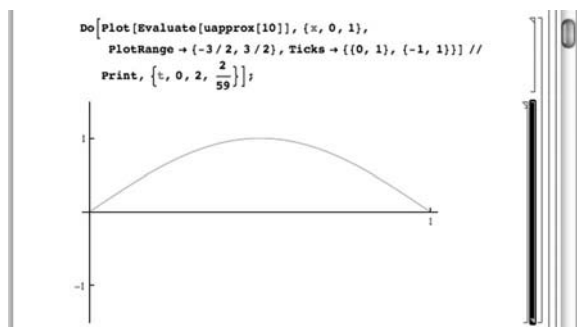
somegraphs = **Table**[**Plot**[**Evaluate**[**uapprox**[10]], { x , 0, 1},
DisplayFunction $\rightarrow$ **Identity**, **PlotRange** $\rightarrow$ $\{-\frac{3}{2}, \frac{3}{2}\}$,
Ticks $\rightarrow$ {{0, 1}, {-1, 1}}, { t , 0, 2, $\frac{2}{15}$ }] ;
toshow = **Partition**[**somegraphs**, 4];
Show[**GraphicsGrid**[**toshow**]]

**FIGURE 6.50**

The motion of the string for 16 equally spaced values of t between 0 and 2

If instead we wished to see the motion of the string, we can use a **Do** loop together with **Print** to generate many graphs and animate the result. We show a frame from the resulting animation.

```
Do[Plot[Evaluate[uapprox[10]], {x, 0, 1},
PlotRange → {-3/2, 3/2}, Ticks → {{0, 1}, {-1, 1}}]//
Print, {t, 0, 2,  $\frac{2}{59}$ }] ;
```



Finally, we remark that **DSolve** can find **D'Alembert's solution** to the wave equation.

```
Clear[u, c]
DSolve[c^2 D[u[x, t], {x, 2}] == D[u[x, t], {t, 2}],
u[x, t], {x, t}]
```

$$\left\{ \left\{ u[x, t] \rightarrow C[1] \left[t - \frac{\sqrt{c^2 x}}{c^2} \right] + C[2] \left[t + \frac{\sqrt{c^2 x}}{c^2} \right] \right\} \right\}$$

$$\text{DSolve} \left[c^2 \partial_{\{x,2\}} u[x, t] == \partial_{\{t,2\}} u[x, t], u[x, t], \{x, t\} \right]$$

$$\left\{ \left\{ u[x, t] \rightarrow C[1] \left[t - \frac{\sqrt{c^2 x}}{c^2} \right] + C[2] \left[t + \frac{\sqrt{c^2 x}}{c^2} \right] \right\} \right\}$$

6.5.2 The Two-Dimensional Wave Equation

One of the more interesting problems involving two spatial dimensions (x and y) is the wave equation. The two-dimensional wave equation in a circular region that is radially symmetric (not dependent on θ) with boundary and initial conditions is expressed in polar coordinates as

$$\begin{cases} c^2 \left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} \right) = \frac{\partial^2 u}{\partial t^2}, & 0 < r < \rho, t > 0 \\ u(\rho, t) = 0, |u(0, t)| < \infty, & t > 0 \\ u(r, 0) = f(r), \frac{\partial u}{\partial t}(r, 0) = g(r), & 0 < r < \rho. \end{cases}$$

Notice that the boundary condition $u(\rho, t) = 0$ indicates that u is fixed at zero around the boundary; the condition $|u(0, t)| < \infty$ indicates that the solution is bounded at the center of the circular region. Like the wave equation discussed previously, this problem is typically solved through separation of variables by assuming a solution of the form $u(r, t) = F(r)G(t)$. Applying separation of variables yields the solution

$$u(r, t) = \sum_{n=1}^{\infty} (A_n \cos ck_n t + B_n \sin ck_n t) J_0(k_n r),$$

α_n represents the n th zero of the Bessel function of the first kind of order zero.

where $\lambda_n = c\alpha_n/\rho$, and the coefficients A_n and B_n are found through application of the initial displacement and velocity functions. With

$$u(r, 0) = \sum_{n=1}^{\infty} A_n J_0(k_n r) = f(r)$$

and the orthogonality conditions of the Bessel functions, we find that

$$A_n = \frac{\int_0^\rho r f(r) J_0(k_n r) dr}{\int_0^\rho r [J_0(k_n r)]^2 dr} = \frac{2}{[J_1(\alpha_n)]^2} \int_0^\rho r f(r) J_0(k_n r) dr, n = 1, 2, \dots$$

Similarly, because

$$\frac{\partial u}{\partial t}(r, 0) = \sum_{n=1}^{\infty} (-ck_n A_n \sin ck_n t + ck_n B_n \cos ck_n t) J_0(k_n r),$$

we have

$$u_t(r, 0) = \sum_{n=1}^{\infty} ck_n B_n J_0(k_n r) = g(r).$$

Therefore,

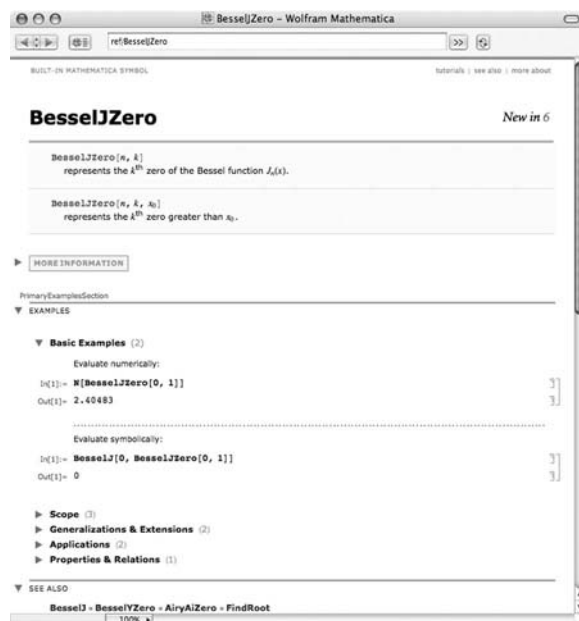
$$B_n = \frac{\int_0^{\rho} rg(r) J_0(k_n r) dr}{ck_n \int_0^{\rho} r [J_0(k_n r)]^2 dr} = \frac{2}{ck_n [J_1(\alpha_n)]^2} \int_0^{\rho} rg(r) J_0(k_n r) dr, \quad n = 1, 2, \dots$$

As a practical matter, in nearly all cases, these formulas are difficult to evaluate.

Example 6.5.2

$$\text{Solve } \begin{cases} \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} = \frac{\partial^2 u}{\partial t^2}, & 0 < r < 1, t > 0 \\ u(1, t) = 0, |u(0, t)| < \infty, & t > 0 \\ u(r, 0) = r(r-1), \frac{\partial u}{\partial t}(r, 0) = \sin \pi r, & 0 < r < 1. \end{cases}$$

Solution In this case, $\rho = 1$, $f(r) = r(r-1)$, and $g(r) = \sin \pi r$. The command `BesselJZero[n,k]` represents the k th zero of the Bessel function $J_n(x)$. To obtain an approximation of the number, use `N`.



$\alpha_{n_} := \text{Evaluate}[\text{BesselJZero}[0, n]/N]$

Next, we define the constants ρ and c and the functions $f(r) = r(r-1)$, $g(r) = \sin \pi r$, and $k_n = \alpha_n/\rho$.

```
c = 1;
ρ = 1;
f[r_] = r(r - 1);
g[r_] = Sin[πr];
kn_ := kn =  $\frac{\alpha_n}{\rho}$ ;
```

The formulas for the coefficients A_n and B_n are then defined so that an approximate solution may be determined. (We use lowercase letters to avoid any possible ambiguity with built-in Mathematica functions.) Note that we use **NIntegrate** to approximate the coefficients and avoid the difficulties in integration associated with the presence of the Bessel function of order zero.

```
an_ := an = (2NIntegrate [rf[r]BesselJ [0, knr] , {r, 0, ρ}]) /
      BesselJ [1, αn]2;
bn_ := bn = (2NIntegrate [rg[r]BesselJ [0, knr] , {r, 0, ρ}]) /
      (c kn BesselJ [1, αn]2)
```

We now compute the first 10 values of A_n and B_n . Because **a** and **b** are defined using the form $a_{n_} := a_x = \dots$ and $b_{n_} := b_x = \dots$, Mathematica remembers these values for later use.

Table[{n, a_n, b_n}, {n, 1, 10}]//TableForm

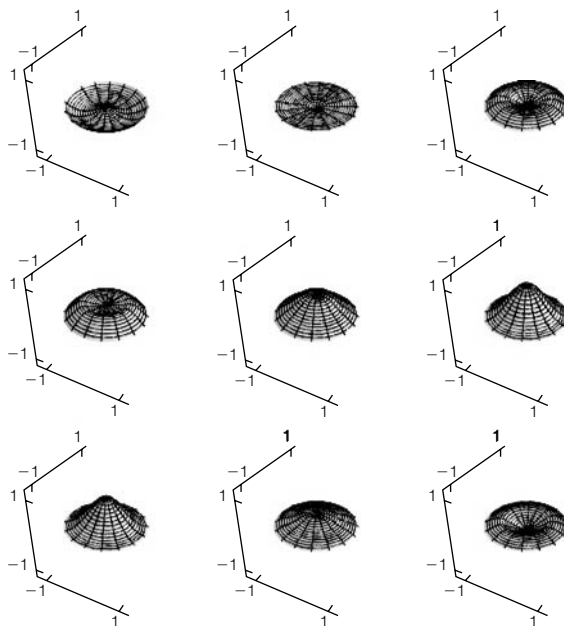
1	-0.323503	0.52118
2	0.208466	-0.145776
3	0.00763767	-0.0134216
4	0.0383536	-0.00832269
5	0.00534454	-0.00250503
6	0.0150378	-0.00208315
7	0.00334937	-0.000882012
8	0.00786698	-0.000814719
9	0.00225748	-0.000410202
10	0.00479521	-0.000399219

The n th term of the series solution is defined in **u**. Then, an approximate solution is obtained in **uapprox** by summing the first 10 terms of **u**.

```
u[n_, r_, t_] := (anCos [cknt] + bnSin [cknt]) BesselJ [0, knr] ;
uapprox[r_, t_] =  $\sum_{n=1}^{10} u[n, r, t]$ ;
```

We graph **uapprox** for several values of t in Figure 6.51.

```
somegraphs =
Table[ParametricPlot3D[{rCos[θ], rSin[θ], uapprox[r, t]},
```

**FIGURE 6.51**

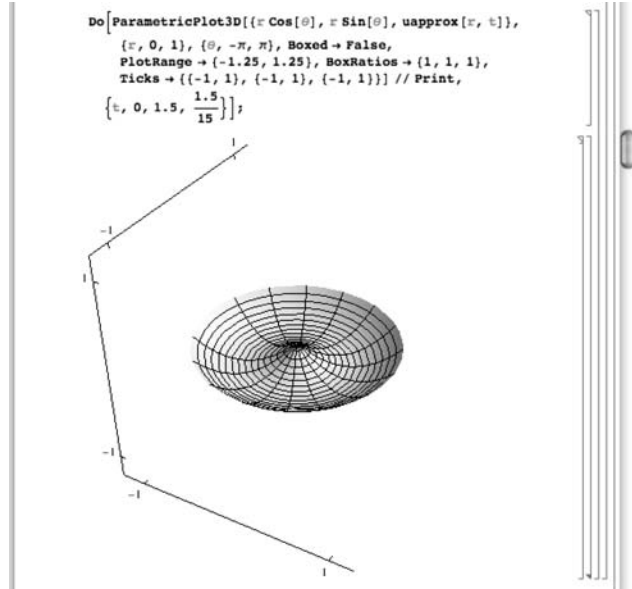
The drumhead for nine equally spaced values of t between 0 and 1.5

```
{r, 0, 1}, {θ, -π, π}, Boxed → False,  
PlotRange → {-1.25, 1.25}, BoxRatios → {1, 1, 1},  
Ticks → {{-1, 1}, {-1, 1}, {-1, 1}}, {t, 0, 1.5,  $\frac{1.5}{8}$ };  
toshow = Partition[somegraphs, 3];  
Show[GraphicsGrid[toshow]]
```

In order to actually watch the drumhead move, we can use a **Do** loop to generate several graphs and animate the result. Be aware, however, that generating many three-dimensional graphics and then animating the results uses a great deal of memory and can take considerable time, even on a relatively powerful computer. We show one frame from the animation that results from the following **Do** loop.

```
Do[ParametricPlot3D[{rCos[θ], rSin[θ], uapprox[r, t]},  
{r, 0, 1}, {θ, -π, π}, Boxed → False,  
PlotRange → {-1.25, 1.25}, BoxRatios → {1, 1, 1},  
Ticks → {{-1, 1}, {-1, 1}, {-1, 1}}//Print,  
{t, 0, 1.5,  $\frac{1.5}{15}$ };
```

If the displacement of the drumhead is not radially symmetric, the problem that describes the displacement of a circular membrane in its general



case is

$$\begin{cases} c^2 \left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} \right) = \frac{\partial^2 u}{\partial t^2}, & 0 < r < \rho, -\pi < \theta < \pi, t > 0 \\ u(\rho, \theta, t) = 0, |u(0, \theta, t)| < \infty, & -\pi \leq \theta \leq \pi, t > 0 \\ u(r, \pi, t) = u(r, -\pi, t), \frac{\partial u}{\partial \theta}(r, \pi, t) = \frac{\partial u}{\partial \theta}(r, -\pi, t), & 0 < r < \rho, t > 0 \\ u(r, \theta, 0) = f(r, \theta), \frac{\partial u}{\partial t}(r, \pi, 0) = g(r, \theta), & 0 < r < \rho, -\pi < \theta < \pi. \end{cases} \quad (6.43)$$

Using separation of variables and assuming that $u(r, \theta, t) = R(r)H(\theta)T(t)$, we obtain that a general solution is given by

$$\begin{aligned} u(r, \theta, t) = & \sum_n a_{0n} J_0(\lambda_{0n} r) \cos(\lambda_{0n} ct) + \sum_{m,n} a_{mn} J_m(\lambda_{mn} r) \cos(m\theta) \cos(\lambda_{mn} ct) \\ & + \sum_{m,n} b_{mn} J_m(\lambda_{mn} r) \sin(m\theta) \cos(\lambda_{mn} ct) + \sum_n A_{0n} J_0(\lambda_{0n} r) \sin(\lambda_{0n} ct) \\ & + \sum_{m,n} A_{mn} J_m(\lambda_{mn} r) \cos(m\theta) \sin(\lambda_{mn} ct) \\ & + \sum_{m,n} B_{mn} J_m(\lambda_{mn} r) \sin(m\theta) \sin(\lambda_{mn} ct), \end{aligned}$$

where J_m represents the m th Bessel function of the first kind, α_{mn} denotes the n th zero of the Bessel function $y = J_m(x)$, and $\lambda_{mn} = \alpha_{mn}/\rho$. The

coefficients are given by the following formulas:

$$\begin{aligned}
 a_{0n} &= \frac{\int_0^{2\pi} \int_0^\rho f(r, \theta) J_0(\lambda_{0n} r) r dr d\theta}{2\pi \int_0^\rho [J_0(\lambda_{0n} r)]^2 r dr} \\
 a_{mn} &= \frac{\int_0^{2\pi} \int_0^\rho f(r, \theta) J_m(\lambda_{mn} r) \cos(m\theta) r dr d\theta}{\pi \int_0^\rho [J_m(\lambda_{mn} r)]^2 r dr} \\
 b_{mn} &= \frac{\int_0^{2\pi} \int_0^\rho f(r, \theta) J_m(\lambda_{mn} r) \sin(m\theta) r dr d\theta}{\pi \int_0^\rho [J_m(\lambda_{mn} r)]^2 r dr} \\
 A_{0n} &= \frac{\int_0^{2\pi} \int_0^\rho g(r, \theta) J_0(\lambda_{0n} r) r dr d\theta}{2\pi \lambda_{0n} c \pi \int_0^\rho [J_0(\lambda_{0n} r)]^2 r dr} \\
 A_{mn} &= \frac{\int_0^{2\pi} \int_0^\rho g(r, \theta) J_m(\lambda_{mn} r) \cos(m\theta) r dr d\theta}{\pi \lambda_{mn} c \int_0^\rho [J_m(\lambda_{mn} r)]^2 r dr} \\
 B_{mn} &= \frac{\int_0^{2\pi} \int_0^\rho g(r, \theta) J_m(\lambda_{mn} r) \sin(m\theta) r dr d\theta}{\pi \lambda_{mn} c \int_0^\rho [J_m(\lambda_{mn} r)]^2 r dr}
 \end{aligned}$$

Example 6.5.3

$$\text{Solve } \begin{cases} 10^2 \left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} \right) = \frac{\partial^2 u}{\partial t^2}, \\ 0 < r < 1, -\pi < \theta < \pi, t > 0 \\ u(1, \theta, t) = 0, |u(0, \theta, t)| < \infty, -\pi \leq \theta \leq \pi, t > 0 \\ u(r, \pi, t) = u(r, -\pi, t), \frac{\partial u}{\partial \theta}(r, \pi, t) = \frac{\partial u}{\partial \theta}(r, -\pi, t), \\ 0 < r < 1, t > 0 \\ u(r, \theta, 0) = \cos(\pi r/2) \sin \theta, \\ \frac{\partial u}{\partial t}(r, \pi, 0) = (r-1) \cos(\pi \theta/2), 0 < r < 1, -\pi < \theta < \pi \end{cases}$$

Solution To calculate the coefficients, we will need to have approximations of the zeros of the Bessel functions, so we use **BesselJZero** together with **N** and **Evaluate** to define α_{mn} to be an approximation of the n th zero of $y = J_m(x)$. We illustrate the use of α_{mn} by using it to compute the first five zeros of $y = J_0(x)$.

$\alpha_{m_}, n_ := \alpha_{m,n} = \text{Evaluate}[\text{BesselJZero}[m, n]/N]$

Table [$\alpha_{0,n}, \{n, 1, 5\}$]

{2.40483, 5.52008, 8.65373, 11.7915, 14.9309}

The appropriate **parameter** values as well as the initial condition functions are defined as follows. Notice that the functions describing the initial displacement

and velocity are defined as the product of functions. This enables the subsequent calculations to be carried out using `NIntegrate`.

```
Clear[a, f, f1, f2, g1, g2, A, c, g, capa, capb, b]
c = 10;
ρ = 1;
f1[r_] = Cos[ $\frac{\pi r}{2}$ ];
f2[θ_] = Sin[θ];
f[r_, θ_] := f[r, θ] = f1[r]f2[θ];
g1[r_] = r - 1;
g2[θ_] = Cos[ $\frac{\pi \theta}{2}$ ];
g[r_, θ_] := g[r, θ] = g1[r]g2[θ];
```

The coefficients a_{0n} are determined with the function `a`.

```
Clear[a]
a[n_] := a[n] =
N[
(NIntegrate[f1[r]BesselJ[0, α0,nr] r, {r, 0, ρ}]
NIntegrate[f2[t], {t, 0, 2π}]) /
(2πNIntegrate[rBesselJ[0, α0,nr]2, {r, 0, ρ}])];
```

Hence, `as` represents a table of the first five values of a_{0n} . `Chop` is used to round off very small numbers to zero.

```
as = Table[a[n]//Chop, {n, 1, 5}]
{0.0, 0, 0, 0}
```

Because the denominator of each integral formula used to find a_{mn} and b_{mn} is the same, the function `bjmn` which computes this value is defined next. A table of nine values of this coefficient is then determined.

```
bjmn[m_, n_] :=
bjmn[m, n] = N[NIntegrate[rBesselJ[m, αm,nr]2, {r, 0, ρ}]]
Table[Chop[bjmn[m, n]], {m, 1, 3}, {n, 1, 3}]

{{0.0811076, 0.0450347, 0.0311763},
{0.0576874, 0.0368243, 0.0270149},
{0.0444835, 0.0311044, 0.0238229}}
```

We also note that in evaluating the numerators of a_{mn} and b_{mn} we must compute $\int_0^\rho r f_1(r) J_m(\alpha_{mn} r) dr$. This integral is defined in `fbjmn` and the corresponding values are found for $n = 1, 2, 3$ and $m = 1, 2, 3$.

```
Clear[fbjmn]
fbjmn[m_, n_] := fbjmn[m, n] =
N[NIntegrate[f1[r]BesselJ[m, αm,nr] r, {r, 0, ρ}]]
```

```

Table[Chop[fbjmn[m, n]], {m, 1, 3}, {n, 1, 3}]
{{0.103574, 0.020514, 0.0103984},
 {0.0790948, 0.0275564, 0.0150381},
 {0.0628926, 0.0290764, 0.0171999}}

```

The formula to compute a_{mn} is then defined and uses the information calculated in `fbjmn` and `bjmn`. As in the previous calculation, the coefficient values for $n = 1, 2, 3$ and $m = 1, 2, 3$ are determined.

```

a[m_, n_] :=
  a[m, n] =
  N[(fbjmn[m, n] NIntegrate[f2[t] Cos[mt], {t, 0, 2π}]) /
    (π bjmn[m, n])];
Table[Chop[a[m, n]], {m, 1, 3}, {n, 1, 3}]
{{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}

```

A similar formula is then defined for the computation of b_{mn} .

```

b[m_, n_] := b[m, n] =
  N[(fbjmn[m, n] NIntegrate[f2[t] Sin[mt], {t, 0, 2π}]) /
    (π bjmn[m, n])];
Table[Chop[b[m, n]], {m, 1, 3}, {n, 1, 3}]
{{1.277, 0.455514, 0.333537}, {0, 0, 0}, {0, 0, 0}}

```

Note that defining the coefficients in this manner `a[m_, n_] := a[m, n] = ...` and `b[m_, n_] := b[m, n] = ...` so that Mathematica “remembers” previously computed values, which reduces computation time. The values of A_{0n} are found similarly to those of a_{0n} . After defining the function `capa` to calculate these coefficients, a table of values is then found.

```

capa[n_] := capa[n] =
  N [(NIntegrate [g1[r] BesselJ [0, α0,n r] r, {r, 0, ρ}]
  NIntegrate[g2[t], {t, 0, 2π}]) /
    (2π α0,n NIntegrate [r BesselJ [0, α0,n r]2, {r, 0, ρ}])];
Table[Chop[capa[n]], {n, 1, 6}]
{0.00142231, 0.0000542518, 0.0000267596, 6.419764234815093*^-6,
 4.958428464118819*^-6, 1.8858472721004333*^-6}

```

The value of the integral of the component of g , `g1`, which depends on r and the appropriate Bessel functions, is defined as `gbjmn`.

```

gbjmn[m_, n_] := gbjmn[m, n] = NIntegrate[g1[r]*
  BesselJ [m, αm,n r] r, {r, 0, ρ}] / N
Table[gbjmn[m, n] // Chop, {m, 1, 3}, {n, 1, 3}]
{{-0.0743906, -0.019491, -0.00989293},
 {-0.0554379, -0.0227976, -0.013039},
 {-0.0433614, -0.0226777, -0.0141684}}

```

Then, A_{mn} is found by taking the product of integrals, $gbjmn$ depending on r and one depending on θ . A table of coefficient values is generated in this case as well.

```
capa[m_, n_] := capa[m, n] =
  N[(gbjmn[m, n] NIntegrate[g2[t] Cos[mt], {t, 0, 2π}]) /
    (π αm,n cbjmn[m, n])];
Table[Chop[capa[m, n]], {m, 1, 3}, {n, 1, 3}]
{{0.0035096, 0.000904517, 0.000457326},
 {-0.00262692, -0.00103252, -0.000583116},
 {-0.000503187, -0.000246002, -0.000150499}}
```

Similarly, the B_{mn} are determined.

```
capb[m_, n_] := capb[m, n] =
  N[(gbjmn[m, n] NIntegrate[g2[t] Sin[mt], {t, 0, 2π}]) /
    (π αm,n cbjmn[m, n])];
Table[Chop[capb[m, n]], {m, 1, 3}, {n, 1, 3}]
{{0.00987945, 0.00254619, 0.00128736},
 {-0.0147894, -0.00581305, -0.00328291},
 {-0.00424938, -0.00207747, -0.00127095}}
```

Now that the necessary coefficients have been found, we construct an approximate solution to the wave equation by using our results. In the following, **term1** represents those terms of the expansion involving a_{0n} , **term2** those terms involving a_{mn} , **term3** those involving b_{mn} , **term4** those involving A_{0n} , **term5** those involving A_{mn} , and **term6** those involving B_{mn} .

```
Clear[term1, term2, term3, term4, term5, term6]
term1[r_, t_, n_] := a[n] BesselJ[0, α0,n r] Cos[α0,n ct];
term2[r_, t_, θ_, m_, n_] :=
  a[m, n] BesselJ[m, αm,n r] Cos[mθ] Cos[αm,n ct];
term3[r_, t_, θ_, m_, n_] :=
  b[m, n] BesselJ[m, αm,n r] Sin[mθ] Cos[αm,n ct];
term4[r_, t_, n_] := capa[n] BesselJ[0, α0,n r] Sin[α0,n ct];
term5[r_, t_, θ_, m_, n_] :=
  capa[m, n] BesselJ[m, αm,n r] Cos[mθ] Sin[αm,n ct];
term6[r_, t_, θ_, m_, n_] :=
  capb[m, n] BesselJ[m, αm,n r] Sin[mθ] Sin[αm,n ct];
```

Therefore, our approximate solution is given as the sum of these terms as computed in u .

```
Clear[u]
u[r_, t_, th_] := Sum5n=1 term1[r, t, n] + Sum3m=1 Sum3n=1 term2[r, t, th, m, n]
```

```

+ Sum_{m=1}^3 Sum_{n=1}^3 term3[r, t, th, m, n] + Sum_{n=1}^5 term4[r, t, n]
+ Sum_{m=1}^3 Sum_{n=1}^3 term5[r, t, th, m, n] + Sum_{m=1}^3 Sum_{n=1}^3 term6[r, t, th, m, n];
uc = Compile[{r, t, th}, u[r, t, th]]

```

```
CompiledFunction[{r, t, th}, u[r, t, th], -CompiledCode-]
```

The solution is *compiled* in `uc`. The command `Compile` is used to compile functions. `Compile` returns a `CompiledFunction` that represents the compiled code. Generally, compiled functions take less time to perform computations than uncompiled functions, although compiled functions can only be evaluated for numerical arguments.

Next, we define the function `tplot`, which uses `ParametricPlot3D` to produce the graph of the solution for a particular value of t . Note that the x and y coordinates are given in terms of polar coordinates.

```

Clear[tplot]
tplot[t_]:=ParametricPlot3D[{rCos[θ], rSin[θ], uc[r, t, θ]},
  {r, 0, 1}, {θ, -π, π}, PlotPoints → {20, 20},
  BoxRatios → {1, 1, 1}, Axes → False, Boxed → False]

```

A table of nine plots for nine equally spaced values of t from $t = 0$ to $t = 1$ using increments of $1/8$ is then generated. This table of graphs is displayed as a graphics array in Figure 6.52.

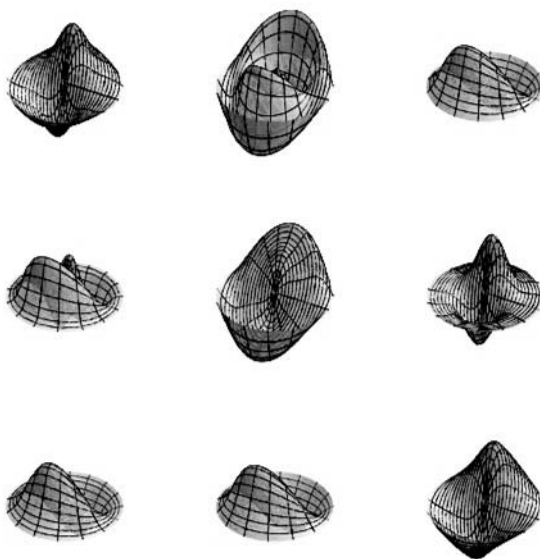


FIGURE 6.52

The drumhead for nine equally spaced values of t from $t = 0$ to $t = 1$

```
somegraphs = Table [tplot[t], {t, 0, 1, 1/8}];
toshow = Partition[somegraphs, 3];
Show[GraphicsGrid[toshow]]
```

Of course, we can generate many graphs with a **Do loop** and animate the result as in the previous example. Be aware, however, that generating many three-dimensional graphics and then animating the results uses a great deal of memory and can take considerable time, even on a relatively powerful computer.

6.5.3 Other Partial Differential Equations

A partial differential equation of the form

$$a(x, y, u) \frac{\partial u}{\partial x} + b(x, y, u) \frac{\partial u}{\partial y} = 0c(x, y, u) \quad (6.44)$$

is called a **first-order, quasilinear partial differential equation**. In the case in which $c(x, y, u) = 0$, equation (6.44) is **homogeneous**; if a and b are independent of u , equation (6.44) is **almost linear**; and when $c(x, y, u)$ can be written in the form $c(x, y, u) = d(x, y)u + s(x, y)$, equation (6.44) is **linear**. Quasilinear partial differential equations can frequently be solved using the *method of characteristics*.

Example 6.5.4 Use the *method of characteristics* to solve the initial-value problem

$$\begin{cases} -3xtu_x + u_t = xt \\ u(x, 0) = x. \end{cases}$$

Solution For this problem, the *characteristic system* is

$$\begin{aligned} \partial x / \partial r &= -3xt, & x(0, s) &= s \\ \partial t / \partial r &= 1, & t(0, s) &= 0 \\ \partial u / \partial r &= xt, & u(0, s) &= s. \end{aligned}$$

We begin by using **DSolve** to solve $\partial t / \partial r = 1$, $t(0, s) = 0$

```
d1 = DSolve[{D[t[r], r]==1, t[0]==0}, t[r], r]
{{t[r] -> r}}
```

and obtain $t = r$. Thus, $\partial x / \partial r = -3xr$, $x(0, s) = s$, which we solve next

```
d2 = DSolve[{D[x[r], r]==-3x[r]r, x[0]==s}, x[r], r]
{{x[r] -> e^(-3/2 r^2) s}}
```

and obtain $x = se^{-3r^2/2}$. Substituting $r = t$ and $x = se^{-3r^2/2}$ into $\partial u / \partial r = xt$, $u(0, s) = s$ and using **DSolve** to solve the resulting equation yields the following result, named **d3**.

$$\mathbf{d3} = \mathbf{DSolve}\left[\left\{\mathbf{D[u[r], r]} == \mathbf{E}^{-\frac{3r^2}{2}} \mathbf{sr}, \mathbf{u[0]} == \mathbf{s}, \mathbf{u[r], r}\right\}\right. \\ \left.\left\{\left\{\mathbf{u[r]} \rightarrow \frac{1}{3} \mathbf{e}^{-\frac{3r^2}{2}} \left(-1 + 4 \mathbf{e}^{\frac{3r^2}{2}}\right) \mathbf{s}\right\}\right\}\right]$$

To find $u(x, t)$, we must solve the system of equations

$$\begin{cases} t = r \\ x = se^{-3r^2/2} \end{cases}$$

for r and s . Substituting $r = t$ into $x = se^{-3r^2/2}$ and solving for s yields $s = xe^{3t^2/2}$. Thus, the solution is given by replacing the values obtained previously in the solution obtained in $\mathbf{d3}$. We do this below by using **ReplaceAll** ($/.$) to replace each occurrence of r and s in $\mathbf{d3}[[1, 1, 2]]$, the solution obtained in $\mathbf{d3}$, by the values $r = t$ and $s = xe^{3t^2/2}$. The resulting output represents the solution to the initial-value problem.

$$\mathbf{d3}[[1, 1, 2]] /. \{r \rightarrow t, s \rightarrow x \mathbf{Exp}[3/2 t^2]\} // \mathbf{Simplify}$$

$$\frac{1}{3} \left(-1 + 4 \mathbf{e}^{\frac{3t^2}{2}}\right) x$$

In this example, **DSolve** can also solve this first-order partial differential equation.

Next, we use **DSolve** to find a general solution of $-3xtu_x + u_t = xt$ and name the resulting output **gensol**.

$$\mathbf{gensol} = \mathbf{DSolve}[-3 \mathbf{x} \mathbf{t} \mathbf{D}[\mathbf{u}[\mathbf{x}, \mathbf{t}], \mathbf{x}] + \mathbf{D}[\mathbf{u}[\mathbf{x}, \mathbf{t}], \mathbf{t}] == \mathbf{x} \mathbf{t}, \\ \mathbf{u}[\mathbf{x}, \mathbf{t}], \{\mathbf{x}, \mathbf{t}\}]$$

$$\left\{\left\{\mathbf{u}[\mathbf{x}, \mathbf{t}] \rightarrow \frac{1}{3} \left(-\mathbf{x} + 3 \mathbf{C}[1] \left[\frac{1}{6} (3 \mathbf{t}^2 + 2 \mathbf{Log}[\mathbf{x}])\right]\right)\right\}\right\}$$

The output

$$\mathbf{C}[1] \left[-\frac{3t^2}{2} - \mathbf{Log}[\mathbf{x}]\right]$$

represents an arbitrary function of $-\frac{3}{2}t^2 - \ln x$. The explicit solution is extracted from **gensol** with **gensol**[[1, 1, 2]], the same way that results are extracted from the output of **DSolve** commands involving ordinary differential equations.

$$\mathbf{gensol}[[1, 1, 2]]$$

$$\frac{1}{3} \left(-\mathbf{x} + 3 \mathbf{C}[1] \left[\frac{1}{6} (3 \mathbf{t}^2 + 2 \mathbf{Log}[\mathbf{x}])\right]\right)$$

To find the solution that satisfies $u(x, 0) = x$, we replace each occurrence of t in the solution by 0.

$$\mathbf{gensol}[[1, 1, 2]] /. \mathbf{t} \rightarrow 0$$

$$\frac{1}{3} \left(-\mathbf{x} + 3 \mathbf{C}[1] \left[\frac{\mathbf{Log}[\mathbf{x}]}{3}\right]\right)$$

Thus, we must find a function $f(x)$ so that

$$-\frac{1}{2}x + f(\ln x) = x$$

$$f(\ln x) = \frac{3}{2}x.$$

Certainly $f(t) = \frac{4}{3}e^{-t}$ satisfies the previous criteria. We define $f(t) = \frac{4}{3}e^{-t}$ and then compute $f(\ln x)$ to verify that $f(\ln x) = \frac{2}{3}x$.

```
Clear[f]
f[t_] = 4Exp[-t]/3;
f[-Log[x]]

$$\frac{4x}{3}$$

```

Thus, the solution to the initial-value problem is given by $-\frac{1}{3}x + f(-\frac{3}{2}t^2 - \ln x)$, which is computed and named `sol`. Of course, the result returned is the same as that obtained previously.

```
sol = Simplify[-x/3 + f[-3t^2/2 - Log[x]]]

$$\frac{1}{3} \left( -1 + 4e^{\frac{3t^2}{2}} \right) x$$

```

Last, we use `Plot3D` to graph `sol` on the rectangle $[0, 20] \times [-2, 2]$ in Figure 6.53.

```
Plot3D[sol, {x, 0, 20}, {t, -2, 2}, PlotRange -> {0, 30},
PlotPoints -> 30, ClippingStyle -> None]
```

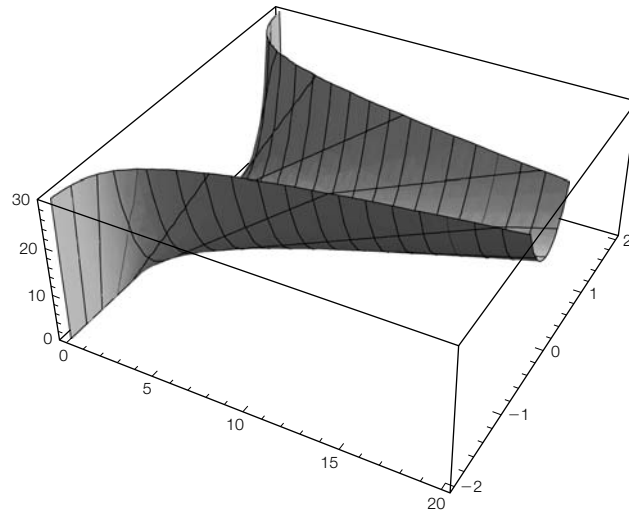


FIGURE 6.53

Plot of $u(x, t) = \frac{1}{3}x \left(4e^{3t^2/2} - 1 \right)$

6.6 EXERCISES

- (a) Solve $(1 + y^2)y' = y \cos x$. (b) Explain the functionality of ProductLog. (c) Show that an implicit solution of the equation is $\frac{1}{2}y^2 + \ln|y| = \sin x + C$. (d) Use ContourPlot to graph various solutions on the rectangle $[0, 10] \times [0, 10]$.
- Solve $xyy' = y^2 - x^2$ and graph several integral curves of the equation. (See Figure 6.54(a).)
- Solve $(-1 + ye^{xy} + y \cos xy) dx + (1 + xe^{xy} + x \cos xy) dy = 0$ and graph several integral curves of the equation. (See Figure 6.54(b).)
- Solve $y' = \sin(2x - y)$, $y(0) = 0.5$. What is the value of $y(1)$? Graph for $0 \leq x \leq 15$.
- Graph the solution of $y' = \sin(ty)$, $y(0) = j$ on $[0, 7]$ for $j = 0.5, 1, \dots, 2.5$.
- Create a Manipulate object that lets you compare the solution of $x'' + ax' + \sin x = 0$ to $x'' + ax' + x = 0$.
- Solve each of the following differential equations or initial-value problems by hand and then verify your results with Mathematica.

(a) $2y'' + 5y' + 5y = 0$, $y(0) = 0$, $y'(0) = 1/2$

(b) $y'' + 4y' + 13y = t \cos^2 3t$

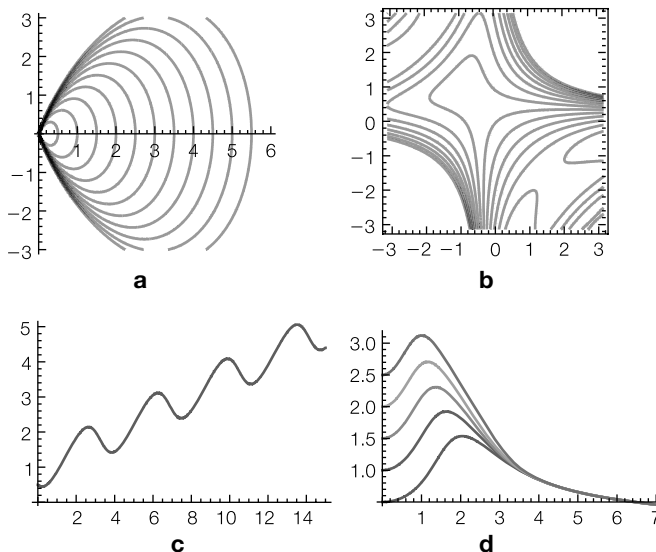


FIGURE 6.54

- (a) Integral curves of $xyy' = y^2 - x^2$. (b) Integral curves of $(-1 + ye^{xy} + y \cos xy) dx + (1 + xe^{xy} + x \cos xy) dy = 0$. (c) The solution of an initial-value problem. (d) Solutions to several initial-value problems

$$\begin{aligned} \text{(c)} \quad & y'' - 2y' + y = e^t \ln t \\ \text{(d)} \quad & t^3 y''' + 16t^2 y'' + 79ty' + 125y = 0 \end{aligned}$$

8. Two lines, l_1 and l_2 , with slopes m_1 and m_2 , respectively, are **orthogonal** (or **perpendicular**) if their slopes satisfy the relationship $m_1 = -1/m_2$. Two curves, C_1 and C_2 , are **orthogonal** (or **perpendicular**) at a point if their respective tangent lines to the curves at that point are perpendicular. Now we want to determine the set of orthogonal curves to a given family of curves. We refer to this set of orthogonal curves as the **family of orthogonal trajectories**. Suppose that a family of curves is defined as $F(x, y) = C$ and that the slope of the tangent line at any point on these curves is $dy/dx = f(x, y)$. Then, the slope of the tangent line on the orthogonal trajectory is $dy/dx = -1/f(x, y)$ so the family of orthogonal trajectories is found by solving the first-order equation $dy/dx = -1/f(x, y)$.

- Determine the family of orthogonal trajectories to the family of curves $y = cx^2$. Confirm your result graphically by graphing members of both families of curves on the same axes.
 - Determine the orthogonal trajectories of the family of curves given by $y^2 - 2cx = c^2$. Graph several members of both families of curves on the same set of axes. Why are these two families of curves said to be **self-orthogonal**?
9. If we are given a family of curves that satisfies the differential equation $dy/dx = f(x, y)$ and we want to find a family of curves that intersects this family at a constant angle θ , we must solve the differential equation

$$\frac{dy}{dx} = \frac{f(x, y) \pm \tan \theta}{1 \mp f(x, y) \tan \theta}.$$

Find a family of curves that intersects the family of curves $x^2 + y^2 = c^2$ at an angle of $\pi/6$. Confirm your result graphically by graphing members of both families of curves on the same axes.

- Find a linear differential equation with general solution $y = c_1 \cos t + c_2 \sin t + e^{t/3}(c_3 \cos 2t + c_4 \sin 2t) + \frac{1}{2}t \sin t$.
- Solve each system and graph various solutions together with the direction field: (a) $\mathbf{X}' = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \mathbf{X}$, (b) $\mathbf{X}' = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{X}$, and (c) $x' = -5x + 3y$, $y' = -2x - 10y$.
- Solve $x' - y = e^{-t}$, $y' + 5x + 2y = \sin 3t$, $x(0) = x_0$, $y(0) = y_0$. Parametrically graph the solution for $(x_0, y_0) = (i, j)$, where i, j take on four equally spaced values between -1 and 1 .

13. Solve $\mathbf{X}' = \begin{pmatrix} -\alpha & \beta \\ -\beta & 0 \end{pmatrix} \mathbf{X}$ if the eigenvalues of the coefficient matrix are (a) real and distinct, (b) real and equal, and (c) complex conjugates. *Hint:* Both DSolve and Assumptions might be helpful.
14. Under certain assumptions, the **FitzHugh–Nagumo equation** that arises in the study of the impulses in a nerve fiber can be written as the system of ordinary differential equations

$$\begin{cases} dV/d\xi = W \\ dW/d\xi = F(V) + R - uW \\ dR/d\xi = \frac{\epsilon}{u}(bR - V - a) \\ V(0) = v_0, W(0) = W_0, R(0) = R_0 \end{cases},$$

where $F(V) = \frac{1}{3}V^3 - V$. (a) Graph the solution to the FitzHugh–Nagumo equation that satisfies the initial conditions $V(0) = 1$, $W(0) = 0$, and $R(0) = 1$ if $\epsilon = 0.08$, $a = 0.7$, $b = 0$, and $u = 1$. (b) Graph the solution that satisfies the initial conditions $V(0) = 1$, $W(0) = 0.5$, and $R(0) = 0.5$ if $\epsilon = 0.08$, $a = 0.7$, $b = 0.8$, and $u = 0.6$.

15. (Controlling the Spread of a Disease).

Sources: Herbert W. Hethcote, “Three basic epidemiological models,” *Applied Mathematical Ecology*, edited by Simon A. Levin, Thomas G. Hallan, and Louis J. Gross, Springer-Verlag (1989), pp. 119–143; Roy M. Anderson and Robert M. May, “Directly transmitted infectious diseases: Control by vaccination,” *Science*, Volume 215, (February 26, 1982), pp. 1053–1060; and J. D. Murray, *Mathematical Biology*, Springer-Verlag (1990), pp. 611–618.

If a person becomes immune to a disease after recovering from it and births and deaths in the population are not taken into account, then the percentage (or proportion) of persons susceptible to becoming infected with the disease, $S(t)$, the percentage of people in the population infected with the disease, $I(t)$, and the percentage of the population recovered and immune to the disease, $R(t)$, can be modeled by the system

$$\begin{cases} S' = -\lambda SI \\ I' = \lambda SI - \gamma I \\ R' = \gamma I \\ S(0) = S_0, I(0) = I_0, R(0) = 0 \end{cases}. \quad (6.45)$$

Because $S(t) + I(t) + R(t) = 1$, once we know $S(t)$ and $I(t)$, we can compute $R(t)$ with $R(t) = 1 - S(t) - I(t)$. This model is called an **SIR model without vital dynamics** because once a person has had the disease, the person becomes immune to the disease, and because births and deaths are not taken into consideration. This model might be used to model diseases that are **epidemic** to a population—those diseases that persist in a population for short periods of time (less than 1 year). Such diseases typically include influenza, measles, rubella, and chickenpox.

If $S_0 < \gamma/\lambda$, $I'(0) = \lambda S_0 I_0 - \gamma I_0 < \lambda \frac{\gamma}{\lambda} I_0 - \gamma I_0 = 0$. Thus, the rate of infection immediately begins to decrease; the disease dies out. On

the other hand, if $S_0 > \gamma/\lambda$, $I'(0) > \lambda S_0 I_0 - \gamma I_0 > 0$, so the rate of infection first increases; an epidemic results.

Although we cannot find explicit formulas for S , I , and R as functions of t , we can, for example, solve for I in terms of S .

- (a) Solve the equation $\frac{dI}{dS} = -\frac{(\lambda S - \gamma)I}{\lambda SI} = -1 + \frac{\rho}{S}$, $\rho = \gamma/\lambda$.

When diseases persist in a population for long periods of time, births and deaths must be taken into consideration. If a person becomes immune to a disease after recovering from it and births and deaths in the population are taken into account, then the percentage of persons susceptible to becoming infected with the disease, $S(t)$, and the percentage of people in the population infected with the disease, $I(t)$, can be modeled by the system

$$\begin{cases} S' = -\lambda SI + \mu - \mu S \\ I' = \lambda SI - \gamma I - \mu I \\ S(0) = S_0, I(0) = I_0 \end{cases}$$

This model is called an **SIR model with vital dynamics** because once a person has had the disease, the person becomes immune to the disease, and because births and deaths are taken into consideration. This model might be used to model diseases that are **endemic** to a population—those diseases that persist in a population for long periods of time (10 or 20 years). Smallpox is an example of a disease that was endemic until it was eliminated in 1977.

- (b) Find and classify the equilibrium points of this system.

Because $S(t) + I(t) + R(t) = 1$, it follows that $S(t) + I(t) \leq 1$. The following table shows the average infectious period, $1/\gamma$, γ , and typical contact numbers, σ , for several diseases during certain epidemics.

Disease	$1/\gamma$	γ	σ
Measles	6.5	0.153846	14.9667
Chickenpox	10.5	0.0952381	11.3
Mumps	19	0.0526316	8.1
Scarlet fever	17.5	0.0571429	8.5

Let us assume that the average lifetime, $1/\mu$, is 70 years so that $\mu = 0.0142857$.

For each of the diseases listed in the previous table, we use the formula $\sigma = \lambda/(\gamma + \mu)$ to calculate the daily contact rate λ .

Disease	λ
Measles	2.51638
Chickenpox	1.23762
Mumps	0.54203
Scarlet fever	0.607143

Diseases such as those listed here can be controlled once an effective and inexpensive vaccine has been developed. Since it is virtually impossible to vaccinate everybody against a disease, we want to know what percentage of a population needs to be vaccinated to eliminate a disease. A population of people has **herd immunity** to a disease if enough people are immune to the disease so that if it is introduced into the population, it will not spread throughout the population. In order to have herd immunity, an infected person must infect less than one uninfected person during the time the person is infectious. Thus, we must have

$$\sigma S < 1.$$

Since $I + S + R = 1$, when $I = 0$ we have that $S = 1 - R$ and, consequently, herd immunity is achieved when

$$\sigma(1 - R) < 1$$

$$\sigma - \sigma R < 1$$

$$-\sigma R < 1 - \sigma$$

$$R > \frac{\sigma - 1}{\sigma} = 1 - \frac{1}{\sigma}.$$

- (c) For each of the diseases listed previously, create a table that estimates the minimum percentage of a population that needs to be vaccinated to achieve herd immunity.
- (d) Using the values in the previous tables, for each disease graph the direction field and several solutions $\begin{cases} S = S(t) \\ I = I(t) \end{cases}$ parametrically.

See texts such as Jordan and Smith's *Nonlinear Ordinary Differential Equations* [23] for discussions of ways to analyze systems such as the Rössler attractor and the Lorenz equations.

16. The **Rössler attractor** is the system

$$\begin{cases} x' = -y - z \\ y' = x + ay \\ z' = bx - cz + xz \end{cases}.$$

Observe that this system is nonlinear because of the product of the x and z terms in the z' equation.

If $a = 0.4$, $b = 0.3$, $x_0 = 1$, $y_0 = 0.4$, and $z(0) = 0.7$, how does the value of c affect solutions to the initial-value problem

$$\begin{cases} x' = -y - z \\ y' = x + ay \\ z' = bx - cz + xz \\ x(0) = x_0, y(0) = y_0, z(0) = z_0 \end{cases} \quad ?$$

Suggestion: Use Manipulate.

17. *Challenge:* Using the linear approximation $\sin \theta = \theta$ for small displacements, derive the equations for a triple pendulum if θ_1 represents the displacement of the upper pendulum (with mass m_1 and length l_1), θ_2 represents the displacement of the upper pendulum (with mass m_2 and length l_2), and θ_3 represents the displacement of the upper pendulum (with mass m_3 and length l_3). Using $g = 32$, illustrate the solution graphically if $m_1 = 3$, $m_2 = 2$, and $m_3 = 1$, $l_1 = 16$, $l_2 = 8$, $l_3 = 16$, $\theta_1(0) = 0$, $\theta_1'(0) = 1$, $\theta_2(0) = 0$, $\theta_2'(0) = 0$, $\theta_3(0) = 0$, and $\theta_3'(0) = -1$.

This page intentionally left blank

References

- [1] Abell, Martha and Braselton, James, *Differential Equations with Mathematica*, Third Edition, Academic Press, 2004.
- [2] Abell, Martha and Braselton, James, *Modern Differential Equations*, Second Edition, Harcourt, 2001.
- [3] Abell, Martha L., Braselton, James P., and Rafter, John A., *Statistics with Mathematica*, Academic Press, 1999.
- [4] Barnsley, Michael, *Fractals Everywhere*, Second Edition, Morgan Kaufmann, 2000.
- [5] Braselton, James P., Abell, Martha L., and Braselton, Lorraine M., "When is a surface *not* orientable?" *International Journal of Mathematical Education in Science and Technology*, Volume 33, Number 4, 2002, pp. 529-541.
- [6] Devaney, Robert L. and Keen, Linda (eds.), *Chaos and Fractals: The Mathematics behind the Computer Graphics*, Proceedings of Symposia in Applied Mathematics, Volume 39, American Mathematical Society, 1989.
- [7] Edwards, Henry C. and Penney, David E., *Calculus with Analytic Geometry*, Fifth Edition, Prentice-Hall, 1998.
- [8] Edwards, Henry C. and Penney, David E., *Differential Equations and Boundary Value Problems: Computing and Modeling*, Third Edition, Pearson/Prentice Hall, 2004.
- [9] Gaylord, Richard J., Kamin, Samuel N., and Wellin, Paul R., *Introduction to Programming with Mathematica*, Second Edition, TELOS/Springer-Verlag, 1996.
- [10] Graff, Karl E., *Wave Motion in Elastic Solids*, Oxford University Press/Dover, 1975/1991.
- [11] Gray, Alfred, Abbena, Elsa, and Salamon, Simon, *Modern Differential Geometry of Curves and Surfaces with Mathematica*, Third Edition, CRC Press, 2006.
- [12] Gray, John W., *Mastering Mathematica: Programming Methods and Applications*, Second Edition, Academic Press, 1997.
- [13] Kyreszig, Erwin, *Advanced Engineering Mathematics*, Seventh Edition, John Wiley & Sons, 1993.
- [14] Larson, Roland E., Hostetler, Robert P., and Edwards, Bruce H., *Calculus with Analytic Geometry*, Sixth Edition, Houghton Mifflin, 1998.
- [15] Maeder, Roman E., *The Mathematica Programmer II*, Academic Press, 1996.
- [16] Maeder, Roman E., *Programming in Mathematica*, Third Edition, Addison-Wesley, 1996.
- [17] Robinson, Clark, *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*, Second Edition, CRC Press, 1999.
- [18] Smith, Hal L. and Waltman, P., *The Theory of the Chemostat: Dynamics of Microbial Competition*, Cambridge University Press, 1995.
- [19] Stewart, James., *Calculus: Concepts and Contexts*, Second Edition, Brooks/Cole, 2001.

- [20] Weisstein, Eric W., *CRC Concise Encyclopedia of Mathematics*, CRC Press, 1999.
- [21] Wolfram, Stephen, *A New Kind of Science*, Wolfram Media, 2002.
- [22] Zwillinger, Daniel, *Handbook of Differential Equations*, Second Edition, Academic Press, 1992.

Index

Symbols

I ., 102, 121, 149, 185, 376

$//N$, 109

I ., 119

$;$, 46

$?$, 17–18, 71–72

$??$, 17–18, 71–72

A

Absolute convergence, 209

Abs[x], 35–36

AlgebraicManipulation, 37, 44

Alternating harmonic series, 209

Alternating series, 209–210

Amortization, 313–314

Animate Selected Graphics, 131

Annuity due, 312

Antiderivatives, 164–166

Apart, 42–43, 204, 437

Append, 283

AppendTo, 283

Apply, 315

Approximate solutions, 110–114

Arc length, 186–190

Arc length function, 375

ArcCosh[x], 35

ArcCos[x], 35–36

ArcCot[x], 35

ArcCsc[x], 35

ArcCsc[x], 35

ArcSec[x], 35

ArcSinh[x], 35

ArcSin[x], 35–36

ArcTanh[x], 35

Area

integrals, 168–174

iterated integrals, 239–240

regions bounded by graph,
180–186

Arithmetic operations, 31–34

Array, 319, 370–371

ArrayPlot, 93–94, 96, 98, 415–416,
420–421, 423–424

Arrow, 6

Arrow, 334–335

AspectRatio, 7, 4, 59, 62, 69, 83,
132, 422

Associated matrix, 358

Assumptions, 41, 552

Astroid, 183–185

$A(t)$, 493

Augmented matrix, 343

Axes, 73–74, 88, 279

AxesLabel, 56, 76–77, 88

AxesOrigin, 73, 142

B

Basic Math Input, 5, 9, 11, 32,
135, 203, 224, 273, 318

Beam problem, 158–161

Bendixson's theorem, 515

Bessel, 295–297

Bessel functions, 274–276, 295,
538–542

BesselJZero, 538–539, 542

Blend, 417

Boxed, 73

BoxRatios, 76, 83, 88–89, 194, 196,
198, 394, 396

Boy surface, 90–92, 116, 431–432

Built-in constants, 34–35

Built-in functions, 35–38

C

Cancel[expression], 42–43, 364

Capacitor, 486–487

Catalan, 34, 116

Cell, 10

CellularAutomaton, 95–100

Characteristic equation, 458, 473

Characteristic polynomial, 358–359,
364, 521–522

Circular plate, 294

Clear[f], 117–119, 127, 135, 138,
151, 218, 229, 269, 273,
280, 288

Clothoid, 68

CMYKColor, 417–418

Coefficient matrix, 342

Cofactor matrix, 328

ColorData, 22, 67

ColorFunction, 67, 69, 73–74, 93,
96, 194, 425–426, 429

ColorSchemes, 22, 55, 129, 250,
417, 425

Column, 137, 167

Column space, 349–351

Compile, 546

CompileFunction, 546

ComplexExpand, 484, 489, 491

Compound interest, 246, 311–312

Conditional command, 119

Conditional convergence, 209

Conic section, 81

Conjugate transpose, 364

Conservative vector field, 380–381

Constant coefficients

n th-order differential equations,
473–475

second-order differential
equations, 458–464

Constants, 34–35

constraints, 198

Continuity, 124–128

Continuum, 126

ContourPlot, 65, 71–73, 76, 79,
104, 108, 111, 113, 116, 139,
222, 227, 229, 231–234, 238,
381–383, 415, 451

ContourPlot3D, 88–89, 105–106

Contours, 73

ContourShading, 73, 111, 142, 451

ContourStyle, 89

Convergence

sequence, 201
series, 203–204, 209
tests, 205–207

ConvertTo, 10

Convolution integral, 486

Convolution theorem, 486–487

Cornu spiral, 68

Cosh[x], 35

Cos[x], 35–36

Cot[x], 35

Cramer's Rule, 471

Create Table/Matrix Palette, 318

Critical number, 148

Critical points, 148, 230–231

Critically damped, 462

Cross product, 331

Cross-Cap, 89

Csc[x], 35

Curl, 382, 389

Curl

vector field, 380

Curvature, 375

Cycloid, 143–145

D

D, 102, 135–138, 224–225, 229,
267, 381

DarkBands, 425

Deferred annuity, 313

- Definite integral, 174–180
- Degenerate critical point, 231
- Denominator[fraction]**, 43, 121, 125
- DensityPlot**, 71–72
- Derivatives
 - antiderivatives, 164–166
 - calculation, 135–138
 - definition, 128–134
 - first derivative test, 148–156
 - mean-value theorem, 146–147
 - multivariable partial and
 - directional derivatives, 224–233
 - second derivative test, 148–156
- Det**, 471
- Differential equations
 - first-order differential equations
 - linear differential equations, 442–450
 - nonlinear differential equations, 450–453
 - numerical solutions, 453–457
 - separable differential equations, 435–442
 - n*th-order differential equations
 - constant coefficients, 473–475
 - Laplace transform, 481–491
 - nonlinear higher-order equations, 492
 - theory, 472–473
 - undetermined coefficients, 475–481
 - partial differential equations
 - first-order quasilinear partial differential equation, 547–549
 - one-dimensional wave equation, 532–537
 - two-dimensional wave equation, 537–547
 - second-order differential equations
 - constant coefficients, 458–464
 - theory, 457–458
 - undetermined coefficients, 464–470
 - variation of parameters, 470–472
 - systems of differential equations
 - homogeneous linear systems, 492–505
 - nonhomogeneous linear systems, 505–532
- Differentiation, *see also* Derivatives
 - antidifferentiation
 - antiderivatives, 164–166
 - u*-substitution, 166–167
 - implicit differentiation, 138–139
 - maximization/minimization problems, 156–164
 - tangent lines, 139–147
- Dirac delta function, 490
- Direction**, 120, 122–124
- Direction field, 17
- Directional derivative, 225–229
- Directory**, 14
- Disease control, 552
- DisplayFunction**, 56, 142
- Div**, 380
- Divergence
 - sequence, 201
 - series, 203–204, 209
 - test, 205
 - vector field, 380
- Divergence theorem, 388–389
- Do**, 195, 293, 504, 536, 540, 547
- Documentation Center**, 2–3, 11, 24–28, 52, 71–72, 95, 339
- Dot product, 331
- Double pendulum, 500–505, 555
- Drawing Tools**, 6, 54
- Drop**, 283
- DSolve**, 16–17, 435–436, 440–441, 445–446, 448–451, 453, 459–460, 464–465, 471, 474, 476, 478, 481, 489, 495–496, 498–499, 501, 508, 537, 547–548, 552
- Dt**, 138
- Dual problem, 368–369
- Duffing's equation, 614–515
- Dynamical system, 264–266, 276, 300–301, 315–316
- Dynamic[x]**, 45, 167
- E**
- e*, 35
- Eigenvalues, 358–361, 366, 430, 518, 521
- Eigenvectors, 358–361, 430
- Eigenystem**, 360–361, 493, 496, 498, 507
- Elementary cellular automaton, 95
- Ellipse, 81
- Ellipsoid, 87
- Elliptical torus, 84–86
- Endemic disease, 553
- Enneper's minimal surface, 250
- Enter**, 5, 26, 32
- Epidemic, 552
- Equation solutions
 - approximate solutions, 110–114
 - exact solutions, 100–109
- Equilibrium point, 514
- EulerGamma**, 34, 115
- Evaluate**, 542
- Exact differential equation, 450–451
- Exact solutions, 100–109
- ExampleData**, 419
- ExpandDenominator[fraction]**, 43
- Expand[expression]**, 39, 41, 162, 363
- ExpandNumerator[fraction]**, 43, 483
- Exponential growth, 445
- Expressions
 - algebraic operations, 39–44
 - defining and evaluating, 47–52
 - naming and evaluating, 44–46
- ExpToTrig**, 115
- Exp[x]**, 35–36
- F**
- Factor**, 459–460, 476
- Factor[expression]**, 39–41, 103, 121, 136, 162, 207, 360–361
- Factorial sequence, 202
- Falling bodies, 447–448
- Family of orthogonal trajectories, 551
- Fibonacci numbers, 431
- Fibonacci sequence, 311
- Filling**, 180
- Finance, 311–315
- FindRoot**, 110–116, 193, 219, 248, 296, 449–450
- First derivative test, 148–156
- First Five Minutes with Mathematica**, 25
- firstguess**, 110
- First-order differential equations
 - linear differential equations, 442–450
 - nonlinear differential equations, 450–453
 - numerical solutions, 453–457
 - separable differential equations, 435–442
- First-order quasilinear partial differential equation, 547–549
- Fit**, 283–284, 286–287
- FitzHugh–Nagumo equation, 552
- Fixed point, 315
- Flatten**, 126, 262–264, 266, 272, 279, 304, 425, 428, 496
- Fobonacci number, 49
- Folium of descartes, 375–380
- Fourier series
 - defining, 287
 - k*th partial sum, 288
 - k*th term, 287
 - one-dimensional heat equation, 290–294
 - partial sums, 288–290

wave equation on circular plate,
294–299

Fraction, 46

Frame, 73–74

Frenet formulas, 405

Frenet frame field, 404

FresnelC, 250, 387

FresnelS, 250, 387

FullSimplify, 208, 217

Fundamental matrix, 493

Fundamental set, 457, 473

Fundamental theorem of calculus,
174

Fundamental theorem of line
integrals, 384

Future value, 312

fvals, 118

ff[x_], 47, 50–51, 147, 169

G

Gabriel's horn, 200–201

Gaussian curvature, 413–414

Gauss–Jordan elimination, 342–349

General form

second-order linear differential
equation, 457

General solution, 473, 492

Generalized Mandelbrot set,
306–307

Globally asymptotically stable
solution, 509

Go, 25

GoldenRatio, 34

Graceful graph, 98

Gradient, 225, 380

GradientFieldPlot3D, 382–383

Gram–Schmidt process, 351–355

Graphics, 6, 54, 131, 278–279, 377

GraphicsArray, 293

GraphicsGrid, 70, 73, 81, 90, 293,
297, 411

Graphics Inspector, 54

GraphicsRow, 86, 88, 130, 139,
278, 280, 304, 334, 421

Graphing

cellular automaton, 95–100

functions of single variable, 52–65

parametric and polar plots, 65–70

parametric curves and surfaces
in space, 82–94

three-dimensional and contour
plots, 71–82

GraphPlot, 98–99

graphs, 81

GrayLevel, 57, 279, 416

Gray's torus, 84–86

GrayTones, 73–74

Green's theorem, 385

Grid, 137, 167, 335–337

Growth constant, 445

H

Harmonic motion, 461–463

Harmonic series, 207

Hearing beats and resonance,
468–469

Help, 24

Help Browser, 40

Herd immunity, 554

Hermite polynomial, 267–269

Hermitian adjoint matrix, 364

Homogeneous linear differential
equation, 442–443

Homogeneous linear systems
differential equations, 492–505

Homogeneous nonlinear differential
equation, 451

Homogeneous n th-order linear
differential equation, 472

Homotopy, 90

Hooke's Law, 461

Hyperbola, 79, 81

Hyperboloid

one sheet, 87

two sheets, 87

I

Identity matrix, 321

IE, 34

ihseq, 79

Ikeda map, 304–305

ImageSize, 427

Implicit differentiation, 138–139

Implicit functions

tangent lines, 141–142

Import, 92, 419

Indeterminate coordinate, 280

Infinite series, 203

Infinity, 34, 117, 122

-Infinity, 117

Inflection points, 148

Information, 82

Input, 318

InputForm, 10, 38

Insert, 321

Inset, 422

Integrals

arc length, 186–190

area, 168–174, 180–186

definite integral, 174–180

iterated integrals, 238–246

solids of revolution, 190–201

vector calculus

line integrals, 384–387

surface integrals, 387–391

Integrate, 68, 165–167, 174–179,

181, 183–184, 186–188, 191,

194, 199, 201, 238–239, 242,

244, 267, 291, 375, 383, 385,

389–391, 437, 446, 451–452, 481

Integration by parts formula, 166

Integrating factor, 444

InterpolatingPolynomial, 286

Interval of convergence, 210

Inverse, 325–326, 338

Inverse functions, 58

Inverse Laplace transform, 481, 485

InverseLaplaceTransform, 481,
483, 485, 488, 490, 502–503

Irrrotational vector field, 380

Iterated integrals, 238–246

J

Jacobian, 514, 520

Join, 328–329, 343

JordanDecomposition, 362–363

Jordan matrix, 361–362

Julia set, 279–282, 299–303, 316

K

Kernel

linear transformation, 355

Klein bottle

orientability, 399–404

Kolmogorov predator–prey
equations, 520

L

Lagrange multiplier, 235, 237

Lagrange's equation, 250

Lagrange's theorem, 236

Laplace transform, 481–491

LaplaceTransform, 481, 483,
487–488, 490, 501–502

Laplacian in polar coordinates, 294

Laplacian of scalar field, 380

leftbox, 168, 170

leftsum, 168–170, 172, 180, 248

Lemniscate of Bernoulli, 185–186

Length, 259, 262, 264, 300, 419,
425, 428

Limit, 46, 117, 119–123, 128–129,
201, 204–206, 208–210

Limit comparison test, 206

Limits

computation, 121–123

continuity, 124–128

functions of two variables,

222–224

graphs and tables in prediction,

117–121

one-sided limits, 123–124

Line, 158, 277, 357
 Linear differential equations, 442–450
 Linear programming
 dual problem, 368–371
 stand form of problem, 366–368
 transportation example, 371–374
 Linear systems of equations
 Gauss–Jordan elimination,
 342–349
 solutions, 337–342
 Linear transformations, 355–358
 Linearly independent differential
 equations, 457, 473
LinearProgramming, 369–370
LinearSolve, 339–342
 Line integrals, 384–387
List, 101, 141
 curve fitting, 283–287
 defining, 251–257
 graphing, 277–282
 manipulation, 269–277
 miscellaneous operations, 283
 nested list, see **Matrix**; **Vector**
 plotting lists of points, 258–269
list, 118, 135, 254, 259–260, 269,
 315, 322
ListContour, 415
ListContourPlot, 422–423, 427–430
ListDensityPlot, 429–430
ListPlot, 61–63, 125–126, 202, 258,
 263–264, 266, 278, 280–281,
 285–286, 300, 415
ListVectorFieldPlot3D, 333
 Locally stable rest point, 514
 Logarithmic integral, 261
 Logistic equation, 438
 Logistic equation with predation,
 454–457
Log[x], 35–36
 Lorenz equations, 530–532
 L–R–C circuit, 486–487

M

Mandelbrot set, 299, 305–311
Manipulate, 6–7, 90–92, 132–134,
 145, 172–173, 214, 216–217,
 248, 251, 257, 335, 378–379,
 441, 463, 467, 497–498,
 505–506, 510, 513, 519, 524,
 529–530, 550, 555
Map, 49, 81, 118, 126, 136–137,
 165, 223–224, 248, 267,
 270–272, 274–276, 278,
 280–282, 304, 308–309, 393,
 456, 496, 504, 526
 Mathematica
 information and help resources,
 1–2, 17–27

 launching, 3–4
 package loading, 13–17
 syntax rules, 13
 user characteristics, 1
 version differences, 2–3
MathSource, 13–14
MathWorld, 14, 18, 29, 94
Matrix
 computations, 325–329
 defining, 317–321
 element extraction, 322–324
 fundamental subspaces, 349–350
 graphical representation, 415–430
 Jordan matrix, 361–362
Matrix, 318, 321
MatrixForm, 319–320, 323–329, 335,
 343–346, 365–366, 480, 521–522
MatrixPlot, 98, 415–417
MatrixPower, 327
 Maximization/minimization problems,
 156–164
Maximize, 152–156, 193, 247,
 369–370
 McLaurin polynomial, 213, 215
 McLaurin series, 213–215
 Mean curvature, 413
 Mean-value theorem
 derivatives, 146–147
Mesh, 76–77, 194
MeshFunctions, 65, 83, 196
middlebox, 168, 171, 173
middlesum, 168–170, 172–173, 180,
 248
 Minimal surface, 249–250
Minimize, 152–156, 193, 247,
 367–369, 373
Miscellaneous, 15
 Möbius strip
 orientability, 396–399
Module, 132
 Monotonic sequence, 201
More Information, 19
 Multivariable calculus
 iterated integrals, 238–246
 limits of functions of two
 variables, 222–224
 partial and directional derivatives,
 224–233

N

Names["form"], 21
NDSolve, 433, 440, 453–454, 511,
 525, 530
Nest, 60, 276, 300–301
 Nested list, see **Matrix**; **Vector**
 Newton's Second Law, 447–448
NIntegrate, 174, 179–180, 183, 188,
 192, 194, 238–239, 378, 543

NMaximize, 193
NMinimize, 193
 Nonhomogeneous linear systems
 differential equations, 505–532
 Nonlinear differential equations,
 450–453
 Nonlinear higher-order differential
 equations, 492
Norm
 integral, 174
 Normal modes, 294
Normalize, 354–355
Norm[v], 330
NRoots, 110, 113, 183
NSolve, 110, 182
 Nullity, 348–349
 Nullspace, 347–350, 356
Numerator[fraction], 43, 121, 125
 Numerical calculations, 31–34
N[%], 199
N[area], 184
N[expression], 109, 113, 159, 207
N[number], 33, 35–36, 538

O

Object, 17
*On Line Encyclopedia of Integer
 Sequences*, 201
 One-dimensional heat equation,
 290–294
 One-dimensional wave equation,
 532–537
 One-sided limits, 123–124
Opacity, 73–74, 86
Options, 17
Options[object], 17–18
 Order preserving path, 391
 Orientable surface, 391–404
 Oriented surface, 388
 Orthogonal curves, 145–146
 Orthogonal lines, 551
Orthogonalize, 354
 Orthonormal vectors, 351
OutputForm, 10
 Outward flux
 vector field, 388–389
 Overdamped, 462
 Overflow error, 280, 305–306

P

Packagename, 14
Packages, 13
Palettes, 11, 22, 44, 318
Panel, 137, 167
 Parabola, 81
 Parallel vectors, 331

- Parametric equations
 - arc length, 187
 - area, 183
 - tangent lines, 143–145
- ParametricPlot**, 20–21, 65–67, 69–70, 78, 116, 144–145, 161, 183, 188, 377, 494, 497, 509, 512–513, 516, 522
- ParametricPlot3D**, 82, 87–88, 90, 191, 196, 198–200, 234, 236–237, 241, 297, 512–513, 546
- Part**, 254, 322, 418–419
- Partial derivative, 224–225
- Partial differential equations
 - first-order quasilinear partial differential equation, 547–549
 - one-dimensional wave equation, 532–537
 - two-dimensional wave equation, 537–547
- Particular solution, 443, 492
- Partition**, 90, 130, 271–272, 275–276, 293, 429, 504
- Pendulum equation with damping, 514
- Permutations**, 81
- π , 35
- Piecewise**, 119
- Play**, 49
- Plot**, 5, 14, 21, 52–54, 57, 62, 64, 79, 103, 120, 149–150, 156, 180, 240, 261, 268, 274, 415, 449–450, 472, 477, 489, 512–513
- Plot3D**, 8, 65, 71, 77, 79, 222, 226, 229, 234, 243, 415, 549
- PlotGradientField**, 227, 229
- PlotJoined**, 278, 285
- PlotLabel**, 56, 69, 73
- PlotPoints**, 70, 73, 76–77, 88, 112, 200
- PlotRange**, 56–57, 62, 70, 83, 132, 149, 196, 226, 377, 394, 396
- PlotStyle**, 7, 14, 55, 57, 67, 69, 73, 86, 144, 195–196, 278, 285
- PlotVectorField**, 456
- Plus**, 315
- Point**, 158, 277, 279, 302
- PointSize**, 62, 280, 285
- PolarPlot**, 65–66, 69–70, 78, 116, 185, 189
- Potential function, 380
- Power series, 210–213
- PowerExpand**, 41–42, 115, 164, 187, 376
- Predator–prey equations, 518–525
- Prepend**, 283
- PrependTo**, 283
- Present value, 312–13
- Prime**, 253, 258
- Prime number theorem, 261
- Principal unit normal vector, 375
- Product**, 315
- Projection
 - vectors, 334, 354–355
- Q**
- QEDecomposition**, 364–365
- Quadric surface, 86–89
- Quit**, 11
- Quit[]**, 11
- R**
- RandomInteger**, 255
- RandomReal**, 49, 118, 222–223, 255
- Range**, 252–253
- Rank, 349
- Ratio test, 205, 210
- RealDigits**, 315
- RealOnly**, 15–16, 28, 34, 63–64, 140, 151, 175–177, 248
- Reduce**, 213, 217, 522
- RegionPlot**, 198, 243
- RegionPlot3D**, 243
- Relative maximum, 148, 230–232
- Relative minimum, 148, 230–232
- ReliefPlot**, 93, 415, 427–430
- Rendering**, 131
- ReplaceAll**, 46, 106, 121, 149, 185, 376
- Rest point, 514
- Return**, 5
- RevolutionPlot3D**, 194
- RGBColor**, 279, 416–417, 424
- Right continuous, 124
- rightbox**, 168, 170
- rightsum**, 168–170, 172, 174, 180, 248
- Roman surface, 90–92
- Root test, 206
- Rössler attractor, 554–555
- Row space, 349
- RowReduce**, 329, 343, 346–350
- RSolve**, 314
- S**
- Saddle point, 231–232
- Save**, 11
- seashell**, 434
- Second derivative test, 148–156, 230
- Second-order differential equations
 - constant coefficients, 458–464
 - theory, 457–458
 - undetermined coefficients, 464–470
 - variation of parameters, 470–472
- Sec[x]**, 35
- Self-orthogonal curves, 551
- Separable differential equations, 435–442
- Sequence, 201–202
- Series**, 215–217
- Series
 - alternating series, 209–210
 - convergence tests, 205–207
 - harmonic series, 207
 - infinite series, 203
 - McLaurin series, 213–215
 - power series, 210–213
 - Taylor series, 213–217
 - Taylor's theorem, 217–220
- Shading**, 76–77
- Short**, 259, 264, 266, 296, 428
- Show**, 53–54, 57–58, 63, 70, 73, 86, 88, 92–94, 130, 185, 278–281, 293, 297, 334–335, 357–358, 527
- Show Changes**, 19
- Show More**, 420, 423
- Sign[x]**, 64
- Simplify**, 32, 37–38, 40–42, 59, 128, 136, 146, 164, 213, 223, 229, 242, 375–376, 381, 385, 387, 471, 478, 480–481, 484–485, 488, 491, 508, 522, 534, 535
- Sine integral function, 166
- Sinh[x]**, 35
- Sin[x]**, 35
- SIR model
 - with vital dynamics, 553
 - without vital dynamics, 552–553
- Slope field, 17
- Smooth curve, 187–189
- SolarColors**, 425
- Solids of revolution
 - surface area, 199–201
 - volume, 190–199
- Solve**, 100–103, 106, 108–109, 112, 138–139, 147, 149, 154, 157, 207, 237, 240, 338–340, 346–347
- Solve**, 437, 451, 453, 457, 459–460, 476–477, 487, 490, 515, 520, 522
- SolveAlways**, 476
- SphericalPlot3D**, 245–246
- Stable fixed point, 315
- Standard form
 - first-order linear differential equation, 442
 - n th-order linear differential equation, 472
 - second-order linear differential equation, 457
- StandardForm**, 10–11
- Standard unit vectors, 331
- Startup Palette**, 64

Stayed-wire problem, 163–164
 Steady-state temperature, 291
 Stokes' theorem, 389
Sum, 204–205, 207, 209–210
 Surface area
 iterated integrals, 239, 241–242
 solids of revolution, 199–201
 Surface integrals, 387–391
 Surface orientability, 391–404
 Syntax rules, 13
 Systems of differential equations
 homogeneous linear systems, 492–505
 nonhomogeneous linear systems, 505–532

T

Table, 48–49, 52, 60–61, 90, 97, 118, 126, 130, 195–196, 202, 272–273, 214, 222–223, 251–253, 255, 257, 260, 262, 265, 267–268, 279, 292–293, 298, 301, 304, 308–310, 319–320, 323, 485, 496, 516–517, 526, 535, 539, 544–545
TableForm, 268–269, 271–274, 289, 292, 335
TableHeadings, 268–269, 273
Take, 259, 324, 351
Talley, 260
 Tangent lines, 139–147
 Tangent plane, 233–234
Tanh[x], 35
Tan[x], 35
 Taylor polynomial, 213, 215
 Taylor series, 213–217
 Taylor's theorem, 217–220
Text, 158, 163–164
Thickness, 86
Thread, 338–339, 348–349
 Threadable functions, 136
time, 157

Together[expression], 40–41, 157, 353
Tooltip, 61, 161, 182, 211, 270
 Tooth surface, 116
toplot, 61, 81, 144, 219, 300
 Torsion, 405
 Torus
 curvature, 414–415
 knot, 84–85, 408–409
 orientability, 392–398
 volume by iterated integral, 245–246
toshow, 293
TraditionalForm, 10, 38–39
Transpose, 323–324, 350, 365
trapezoid, 248
TreeForm, 39
TreePlot, 98–100
 Trefoil knot, 411–414
TrigExpand, 37, 103, 115
TrigReduce, 37
TrigToExp, 38
 Triple iterated integrals, 244–246
tubeplot, 409
 Two-dimensional wave equation, 537–547

U

Umbilic torus, 82–83, 432
 Underdamped, 462
 Underflow error, 280
 Undetermined coefficients
 n th-order differential equations, 475–481
 second-order differential equations, 464–470
Union, 96, 526
 Unit binormal vector field, 405
 Unit circle, 66, 77–79
 Unit normal field, 413
 Unit normal vector field, 404–405
 Unit tangent vector, 375
 Unit tangent vector field, 404–407

Unitary matrix, 364
 Unstable fixed point, 315
 Unstable node, 525
 Unstable rest point, 514
 Unstable spiral, 525
 u -substitution, 166–167

V

Van der Pol's equation, 511, 525–530
 Vector
 calculus
 line integrals, 384–387
 nonorientability, 391–404
 surface integrals, 387–391
 tangents, normals, and curvature in $\mathbf{R}^3$, 404–415
 vector-valued functions, 374–384
 computations
 basic operations, 329–330
 projection, 334–337
 vectors in 3-space, 330–333
 defining, 321–322
 Vector triple product, 413
VectorAnalysis, 380, 382, 389
VectorFieldPlot, 381, 438, 494, 516–519
VectorFieldPlots, 15–17, 28, 227, 229–230, 381–383, 393
 Verhuist equation, 438
\$VersionNumber, 2
VertexLabeling, 100
ViewPoint, 76–77
 Volume
 iterated integrals, 239, 243–244
 solids of revolution, 190–199

W

Wave equation on circular plate, 294–299
 Wronskian, 457, 473, 480