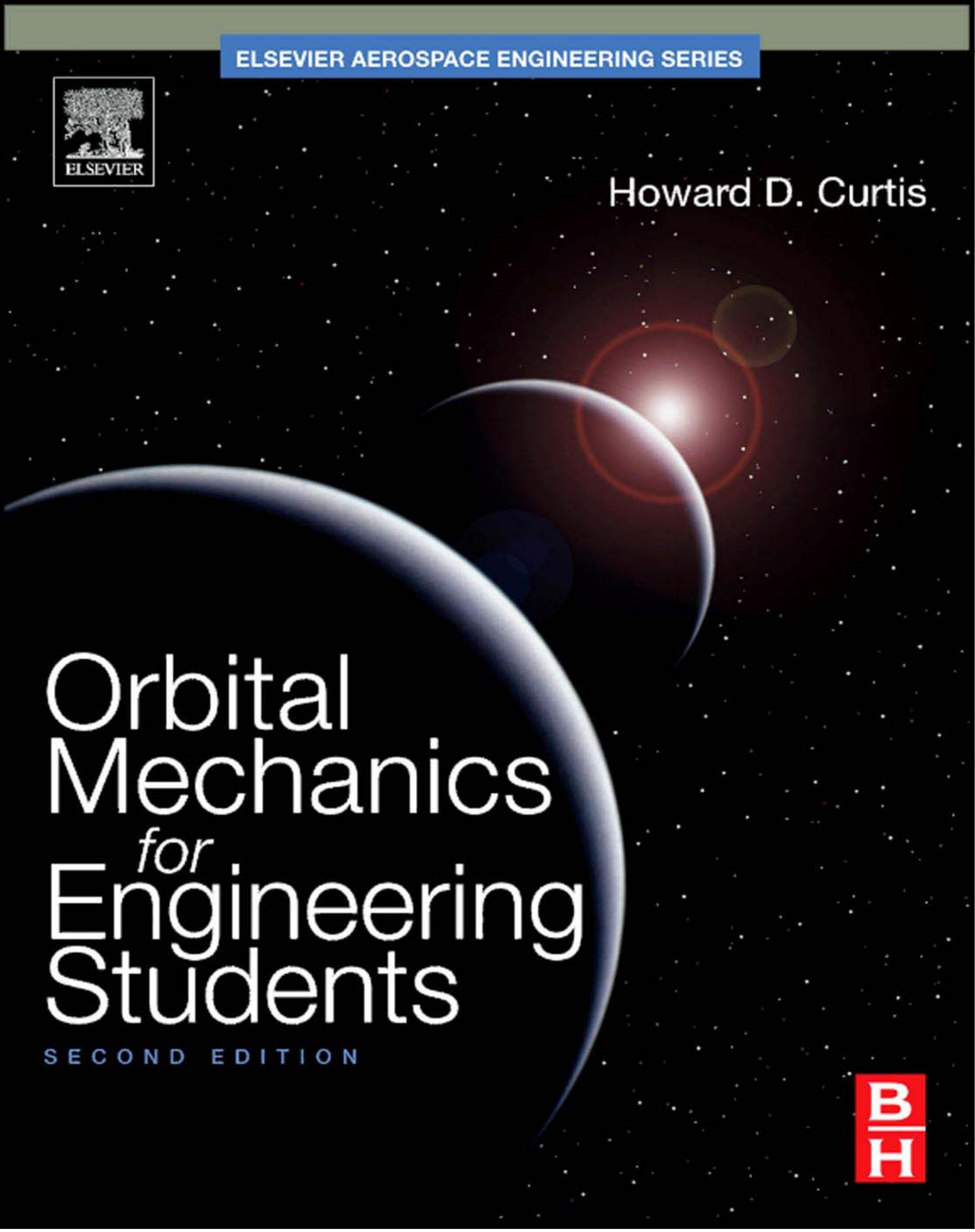
ELSEVIER AEROSPACE ENGINEERING SERIES
ELSEVIER
Howard D. Curtis
Orbital Mechanics for Engineering Students
SECOND EDITION
B H

Orbital Mechanics for Engineering Students

This page intentionally left blank

Orbital Mechanics for Engineering Students

Second Edition

Howard D. Curtis
Professor of Aerospace Engineering
Embry-Riddle Aeronautical University
Daytona Beach, Florida

ELSEVIER
AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Butterworth-Heinemann is an imprint of Elsevier

Butterworth-Heinemann is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

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: 978-0-12-374778-5 (Case bound)
ISBN: 978-1-85617-954-6 (Case bound with on line testing)

For information on all Butterworth–Heinemann publications
visit our Web site at www.elsevierdirect.com

Printed in the United States of America
09 10 11 12 13 10 9 8 7 6 5 4 3 2 1

To my parents, Rondo and Geraldine.

This page intentionally left blank

Contents

Preface

The purpose of this book, like the first edition, is to provide an introduction to space mechanics for undergraduate engineering students. It is not directed towards graduate students, researchers and experienced practitioners, who may nevertheless find useful review material within the book's contents. The intended readers are those who are studying the subject for the first time and have completed courses in physics, dynamics and mathematics through differential equations and applied linear algebra. I have tried my best to make the text readable and understandable to that audience. In pursuit of that objective I have included a large number of example problems that are explained and solved in detail. Their purpose is not to overwhelm but to elucidate. I find that students like the "teach by example" method. I always assume that the material is being seen for the first time and, wherever possible, I provide solution details so as to leave little to the reader's imagination. The numerous figures throughout the book are also intended to aid comprehension. All of the more labor-intensive computational procedures are implemented in MATLAB® code.

CHANGES TO THE SECOND EDITION

Most of the content and style of the first edition has been retained. Some topics have been revised, rearranged or relocated. I have corrected all of the errors that I discovered or that were reported to me by students, teachers, reviewers and other readers. Key terms are now listed at the end of each chapter. The answers in the example problems are boxed instead of underlined. The homework problems at the end of each chapter have been grouped by applicable section. There are many new example problems and homework problems.

Chapter 1, which is a review of particle dynamics, begins with a new section on vectors, which are used throughout the book. Therefore, I thought a brief review of basic vector concepts and operations was appropriate. The chapter concludes with a new section on the numerical integration of ordinary differential equations (ODEs). These Runge-Kutta and predictor-corrector methods, which I implemented in the MATLAB codes *rk1_4.m*, *rkf45.m* and *heun.m*, facilitate the investigation and simulation of space mechanics problems for which analytical, closed-form solutions are not available. Many of the book's new example problems illustrate applications of this kind. Throughout the text I mostly use the ODE solvers *heun.m* (fixed time step) and *rkf45.m* (variable time step) because they work well and the scripts (see Appendix D) are short and easy to read. In every case I checked their results against two of MATLAB's own suite of ODE solvers, primarily *ode23.m* and *ode45.m*. These general-purpose codes are far more elegant (and lengthy) than the ones mentioned above. They may be listed by issuing the MATLAB `type` command.

I have added two algorithms to Chapter 2 for numerically integrating the two-body equations of motion: an algorithm for propagating a state vector as a function of true anomaly, and an algorithm for finding the roots of a function by the bisection method. The last one is useful for determining the Lagrange points in the restricted three-body problem.

Chapter 4 now includes the material on coordinate transformations previously found in this and other chapters. Section 4.5 includes a more general treatment of the Euler elementary rotation sequences, with emphasis on the classical (3-1-3) Euler sequence and the yaw-pitch-roll (3-2-1) sequence. Algorithms were added to calculate the right ascension and declination from the position vector and to calculate the classical Euler angles and the yaw, pitch and roll angles from the direction cosine matrix. I also moved all discussion of ground tracks into Chapter 4 and offer an algorithm for obtaining the ground track of a satellite from its orbital elements.

Chapter 6 concludes with a new section on nonimpulsive (finite burn time) orbital change maneuvers, including MATLAB simulations.

Chapter 7 now includes an algorithm to find the position, velocity and acceleration of a spacecraft relative to an LVLH frame. Also new to this chapter is the derivation of the linearized equations of relative motion for an elliptical (not necessarily circular) reference orbit.

New to Chapter 9 is a discussion of quaternions and associated algorithms for use in numerically solving Euler's equations of rigid body motion to obtain the evolution of spacecraft attitude. Quaternions can be used with MATLAB's `rotate` command to produce simple animations of spacecraft motion.

Appendices C and D have changed. The MATLAB script in Appendix C was revised. Appendix D no longer contains the listings of MATLAB codes. Instead, the algorithms are listed along with the world wide web addresses from which they may be downloaded. This edition contains over twice the number of MATLAB M-files as did the first.

ORGANIZATION

The organization of the book remains the same as that of the first edition. Chapter 1 is a review of vector kinematics in three dimensions and of Newton's laws of motion and gravitation. It also focuses on the issue of relative motion, crucial to the topics of rendezvous and satellite attitude dynamics. The new material on ordinary differential equation solvers will be useful for students who are expected to code numerical simulations in MATLAB or other programming languages. Chapter 2 presents the vector-based solution of the classical two-body problem, resulting in a host of practical formulas for the analysis of orbits and trajectories of elliptical, parabolic and hyperbolic shape. The restricted three-body problem is covered in order to introduce the notion of Lagrange points and to present the numerical solution of a lunar trajectory problem. Chapter 3 derives Kepler's equations, which relate position to time for the different kinds of orbits. The universal variable formulation is also presented. Chapter 4 is devoted to describing orbits in three dimensions. Coordinate transformations and the Euler elementary rotation sequences are defined. Procedures for transforming back and forth between the state vector and the classical orbital elements are addressed. The effect of the earth's oblateness on the motion of an orbit's ascending node and eccentricity vector is examined. Chapter 5 is an introduction to preliminary orbit determination, including Gibbs's and Gauss's methods and the solution of Lambert's problem. Auxiliary topics include topocentric coordinate systems, Julian day numbering and sidereal time. Chapter 6 presents the common means of transferring from one orbit to another by impulsive delta-v maneuvers, including Hohmann transfers, phasing orbits and plane changes. Chapter 7 is a brief introduction to relative motion in general and to the two-impulse rendezvous problem in particular. The latter is analyzed using the Clohessy-Wiltshire equations, which are derived in this chapter. Chapter 8 is an introduction to interplanetary mission design using patched conics. Chapter 9 presents those elements of rigid-body dynamics required to characterize the attitude of a space vehicle. Euler's equations of rotational motion are derived and applied in a number of example problems. Euler angles, yaw-pitch-roll angles and quaternions are presented as ways to describe the attitude of rigid body. Chapter 10 describes the methods of controlling, changing and stabilizing the attitude of spacecraft by means of thrusters, gyros and other devices. Finally, Chapter 11 is a brief introduction to the characteristics and design of multi-stage launch vehicles.

Chapters 1 through 4 form the core of a first orbital mechanics course. The time devoted to Chapter 1 depends on the background of the student. It might be surveyed briefly and used thereafter simply as a reference. What follows Chapter 4 depends on the objectives of the course.

Chapters 5 through 8 carry on with the subject of orbital mechanics. Chapter 6 on orbital maneuvers should be included in any case. Coverage of Chapters 5, 7 and 8 is optional. However, if all of Chapter 8 on interplanetary missions is to form a part of the course, then the solution of Lambert's problem (Section 5.3) must be studied beforehand.

Chapters 9 and 10 must be covered if the course objectives include an introduction to spacecraft dynamics. In that case Chapters 5, 7 and 8 would probably not be covered in depth.

Chapter 11 is optional if the engineering curriculum requires a separate course in propulsion, including rocket dynamics.

The important topic of spacecraft control systems is omitted. However, the material in this book and a course in control theory provide the basis for the study of spacecraft attitude control.

To understand the material and to solve problems requires using a lot of undergraduate mathematics. Mathematics, of course, is the language of engineering. Students must not forget that Sir Isaac Newton had to invent calculus so he could solve orbital mechanics problems in more than just a heuristic way. Newton (1642–1727) was an English physicist and mathematician, whose 1687 publication *Mathematical Principles of Natural Philosophy* ("the *Principia*") is one of the most influential scientific works of all time. It must be noted that the German mathematician Gottfried Wilhelm von Leibnitz (1646–1716), is credited with inventing infinitesimal calculus independently of Newton in the 1670s.

In addition to honing their math skills, students are urged to take advantage of computers (which, incidentally, use the binary numeral system developed by Leibnitz). There are many commercially available mathematics software packages for personal computers. Wherever possible they should be used to relieve the burden of repetitive and tedious calculations. Computer programming skills can and should be put to good use in the study of orbital mechanics. The elementary MATLAB programs referred to in Appendix D of this book illustrate how many of the procedures developed in the text can be implemented in software. All of the scripts were developed and tested using MATLAB version 7.7. Information about MATLAB, which is a registered trademark of The MathWorks, Inc., may be obtained from

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA, 01760-2089, USA
www.mathworks.com

Appendix A presents some tables of physical data and conversion factors. Appendix B is a road map through the first three chapters, showing how the most fundamental equations of orbital mechanics are related. Appendix C shows how to set up the n-body equations of motion and program them in MATLAB. Appendix D contains the web locations of the M-files of all of the MATLAB-implemented algorithms and example problems presented in the text. Appendix E shows that the gravitational field of a spherically symmetric body is the same as if the mass were concentrated at its center.

The field of astronautics is rich and vast. References cited throughout this text are listed at the end of the book. Also listed are other books on the subject that might be of interest to those seeking additional insights.

SUPPLEMENTS TO THE TEXT

For purchasers of this book:

Copies of the MATLAB M-files listed in Appendix D can be freely downloaded from the companion website accompanying this book. To access these files please visit www.elsevierdirect.com/9780123747785 and click on the "companion site" link.

For instructors using this book as text for their course:

Please visit www.textbooks.elsevier.com to register for access to the solutions manual, PowerPoint® lecture slides and other resources.

This page intentionally left blank

Acknowledgments

Since the publication of the first edition and during the preparation of this one, I have received helpful criticism, suggestions and advice from many sources locally and worldwide. I thank them all and regret that time and space limitations prohibited the inclusion of some recommended additional topics that would have enhanced the book. I am especially indebted to those who reviewed the proposed revision plan and second edition manuscript for the publisher for their many suggestions on how the book could be improved. Thanks to:

Rodney Anderson	University of Colorado at Boulder
Dale Chimenti	Iowa State University
David Cicci	Auburn University
Michael Freeman	University of Alabama
William Garrard	University of Minnesota
Peter Ganatos	City College of New York
Liam Healy	University of Maryland
Sanjay Jayaram	St. Louis University
Colin McInnes	University of Strathclyde
Eric Mehiel	Cal Poly, San Luis Obispo
Daniele Mortari	Texas A&M University
Roy Myose	Wichita State University
Steven Nerem	University of Colorado
Gianmarco Radice	University of Glasgow
Alistair Revell	University of Manchester
Trevor Sorensen	University of Kansas
David Spencer	Penn State University
Rama K. Yedavalli	Ohio State University

It has been a pleasure to work with the people at Elsevier, in particular Joseph P. Hayton, Publisher, Maria Alonso, Assistant Editor, and Anne B. McGee, Project Manager. I appreciate their enthusiasm for the book, their confidence in me, and all the work they did to move this project to completion.

Finally and most importantly, I must acknowledge the patience and support of my wife, Mary, who was a continuous source of optimism and encouragement throughout the yearlong revision effort.

Howard D. Curtis
Embry-Riddle Aeronautical University
Daytona Beach, Florida

CHAPTER

Dynamics of point masses

1

Chapter outline

1.1 INTRODUCTION

This chapter serves as a self-contained reference on the kinematics and dynamics of point masses as well as some basic vector operations and numerical integration methods. The notation and concepts summarized here will be used in the following chapters. Those familiar with the vector-based dynamics of particles can simply page through the chapter and then refer back to it later as necessary. Those who need a bit more in the way of review will find the chapter contains all of the material they need in order to follow the development of orbital mechanics topics in the upcoming chapters.

We begin with a review of vectors and some vector operations after which we proceed to the problem of describing the curvilinear motion of particles in three dimensions. The concepts of force and mass are considered next, along with Newton's inverse-square law of gravitation. This is followed by a presentation of Newton's second law of motion ("force equals mass times acceleration") and the important concept of angular momentum.

As a prelude to describing motion relative to moving frames of reference, we develop formulas for calculating the time derivatives of moving vectors. These are applied to the computation of relative velocity and acceleration. Example problems illustrate the use of these results, as does a detailed consideration of how the earth's rotation and curvature influence our measurements of velocity and acceleration. This brings in the curious concept of Coriolis force. Embedded in exercises at the end of the chapter is practice in verifying several fundamental vector identities that will be employed frequently throughout the book.

The chapter concludes with an introduction to numerical integration methods, which can be called upon to solve the equations of motion when an analytical solution is not possible.

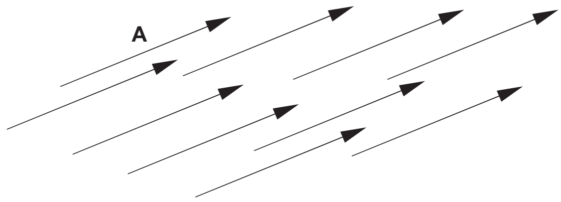

FIGURE 1.1

All of these vectors may be denoted **A**, since their magnitudes and directions are the same.

1.2 VECTORS

A vector is an object that is specified by both a magnitude and a direction. We represent a vector graphically by a directed line segment, that is, an arrow pointing in the direction of the vector. The end opposite the arrow is called the tail. The length of the arrow is proportional to the magnitude of the vector. Velocity is a good example of a vector. We say that a car is traveling east at eighty kilometers per hour. The direction is east and the magnitude, or speed, is 80 km/h. We will use boldface type to represent vector quantities and plain type to denote scalars. Thus, whereas B is a scalar, $\mathbf{B}$ is a vector.

Observe that a vector is specified solely by its magnitude and direction. If $\mathbf{A}$ is a vector, then all vectors having the same physical dimensions, the same length and pointing in the same direction as $\mathbf{A}$ are denoted $\mathbf{A}$, regardless of their line of action, as illustrated in Figure 1.1. Shifting a vector parallel to itself does not mathematically change the vector. However, parallel shift of a vector might produce a different physical effect. For example, an upward 5 kN load (force vector) applied to the tip of an airplane wing gives rise to quite a different stress and deflection pattern in the wing than the same load acting at the wing's mid-span.

The magnitude of a vector $\mathbf{A}$ is denoted $\|\mathbf{A}\|$, or, simply A.

Multiplying a vector $\mathbf{B}$ by the reciprocal of its magnitude produces a vector which points in the direction of $\mathbf{B}$, but it is dimensionless and has a magnitude of one. Vectors having unit dimensionless magnitude are called unit vector*s*. We put a hat (^) over the letter representing a unit vector. Then we can tell simply by inspection that, for example, $\hat{\mathbf{u}}$ is a unit vector, as are $\hat{\mathbf{B}}$ and $\hat{\mathbf{e}}$.

It is convenient to denote the unit vector in the direction of the vector $\mathbf{A}$ as $\hat{\mathbf{u}}_A$. As pointed out above, we obtain this vector from $\mathbf{A}$ as follows

$$\hat{\mathbf{u}}_A = \frac{\mathbf{A}}{A} \tag{1.1}$$

Likewise, $\hat{\mathbf{u}}_C = \mathbf{C}/C$, $\hat{\mathbf{u}}_F = \mathbf{F}/F$, etc.

The sum or ***resultant*** of two vectors is defined by the parallelogram rule (Figure 1.2). Let $\mathbf{C}$ be the sum of the two vectors $\mathbf{A}$ and $\mathbf{B}$. To form that sum using the parallelogram rule, the vectors $\mathbf{A}$ and $\mathbf{B}$ are shifted parallel to themselves (leaving them unaltered) until the tail of $\mathbf{A}$ touches the tail of $\mathbf{B}$. Drawing dotted lines through the head of each vector parallel to the other completes a parallelogram. The diagonal from the tails of $\mathbf{A}$ and $\mathbf{B}$ to the opposite corner is the resultant $\mathbf{C}$. By construction, vector addition is commutative, that is,

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A} \tag{1.2}$$

A ***Cartesian coordinate system*** in three dimensions consists of three axes, labeled x, y and z, which intersect at the origin O. We will always use a right-handed Cartesian coordinate system, which means if you wrap the fingers of your right hand around the z axis, with the thumb pointing in the positive z direction,

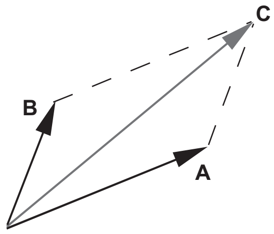

FIGURE 1.2
Parallelogram rule of vector addition.

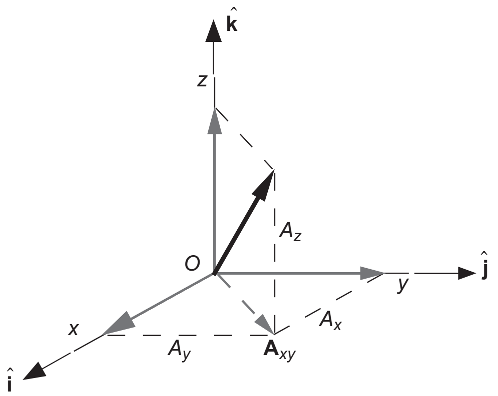

FIGURE 1.3
Three-dimensional, right-handed Cartesian coordinate system.

your fingers will be directed from the *x* axis towards the *y* axis. Figure 1.3 illustrates such a system. Note that the unit vectors along the *x*, *y* and *z*-axes are, respectively, $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$.

In terms of its Cartesian components, and in accordance with the above summation rule, a vector **A** is written in terms of its components A_x, A_y and A_z as

$$\mathbf{A} = A_x\hat{\mathbf{i}} + A_y\hat{\mathbf{j}} + A_z\hat{\mathbf{k}} \tag{1.3}$$

The projection of **A** on the *xy* plane is denoted $\mathbf{A}_{xy}$. It follows that

$$\mathbf{A}_{xy} = A_x\hat{\mathbf{i}} + A_y\hat{\mathbf{j}}$$

According to the Pythagorean theorem, the magnitude of **A** in terms of its Cartesian components is

$$A = \sqrt{A_x^{\,2} + A_y^{\,2} + A_z^{\,2}} \tag{1.4}$$

From Equations 1.1 and 1.3, the unit vector in the direction of **A** is

$$\hat{\mathbf{u}}_{\mathrm{A}} = \cos\theta_x\hat{\mathbf{i}} + \cos\theta_y\hat{\mathbf{j}} + \cos\theta_z\hat{\mathbf{k}} \tag{1.5}$$

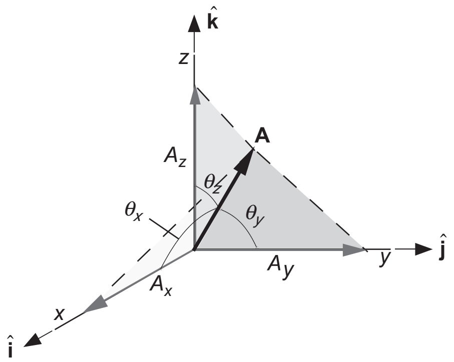

FIGURE 1.4

Direction angles in three dimensions.

where

$$\cos\theta_x = \frac{A_x}{A} \qquad \cos\theta_y = \frac{A_y}{A} \qquad \cos\theta_z = \frac{A_z}{A} \tag{1.6}$$

The direction angles θ_x, θ_y and θ_z are illustrated in Figure 1.4, and are measured between the vector and the positive coordinate axes. Note carefully that the sum of θ_x, θ_y and θ_z is not in general known a priori and cannot be assumed to be, say, 180 degrees.

Example 1.1

Calculate the direction angles of the vector $\mathbf{A} = \hat{\mathbf{i}} - 4\hat{\mathbf{j}} + 8\hat{\mathbf{k}}$.

Solution

First, compute the magnitude of **A** by means of Equation 1.4:

$$A = \sqrt{1^2 + (-4)^2 + 8^2} = 9$$

Then Equations 1.6 yield

$$\theta_x = \cos^{-1}\left(\frac{A_x}{A}\right) = \cos^{-1}\left(\frac{1}{9}\right) \Rightarrow \boxed{\theta_x = 83.62^\circ}$$

$$\theta_y = \cos^{-1}\left(\frac{A_y}{A}\right) = \cos^{-1}\left(\frac{-4}{9}\right) \Rightarrow \boxed{\theta_y = 116.4^\circ}$$

$$\theta_z = \cos^{-1}\left(\frac{A_z}{A}\right) = \cos^{-1}\left(\frac{8}{9}\right) \Rightarrow \boxed{\theta_z = 27.27^\circ}$$

Observe that $\theta_x + \theta_y + \theta_z = 227.3^\circ$.

Multiplication and division of two vectors are undefined operations. There are no rules for computing the product **AB** and the ratio **A/B**. However, there are two well-known binary operations on vectors: the dot product and the cross product. The ***dot product*** of two vectors is a scalar defined as follows,

$$\mathbf{A}\cdot\mathbf{B} = AB\cos\theta \tag{1.7}$$

where θ is the angle between the heads of the two vectors, as shown in Figure 1.5. Clearly,

$$\mathbf{A}\cdot\mathbf{B} = \mathbf{B}\cdot\mathbf{A} \tag{1.8}$$

If two vectors are perpendicular to each other, then the angle between them is 90°. It follows from Equation 1.7 that their dot product is zero. Since the unit vectors $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ of a Cartesian coordinate system are mutually orthogonal and of magnitude one, Equation 1.7 implies that

$$\begin{aligned}\hat{\mathbf{i}}\cdot\hat{\mathbf{i}} &= \hat{\mathbf{j}}\cdot\hat{\mathbf{j}} = \hat{\mathbf{k}}\cdot\hat{\mathbf{k}} = 1\\ \hat{\mathbf{i}}\cdot\hat{\mathbf{j}} &= \hat{\mathbf{i}}\cdot\hat{\mathbf{k}} = \hat{\mathbf{j}}\cdot\hat{\mathbf{k}} = 0\end{aligned} \tag{1.9}$$

Using these properties it is easy to show that the dot product of the vectors **A** and **B** may be found in terms of their Cartesian components as

$$\mathbf{A}\cdot\mathbf{B} = A_xB_x + A_yB_y + A_zB_z \tag{1.10}$$

If we set $\mathbf{B} = \mathbf{A}$, then it follows from Equations 1.4 and 1.10 that

$$A = \sqrt{\mathbf{A}\cdot\mathbf{A}} \tag{1.11}$$

The dot product operation is used to project one vector onto the line of action of another. We can imagine bringing the vectors tail to tail for this operation, as illustrated in Figure 1.6. If we drop a perpendicular

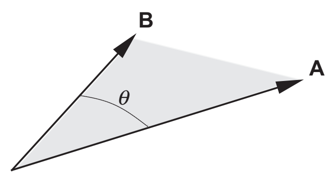

FIGURE 1.5

The angle between two vectors brought tail to tail by parallel shift.

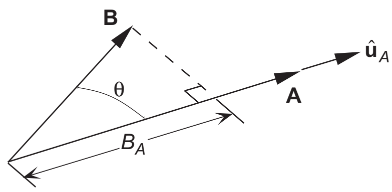

FIGURE 1.6

Projecting the vector **B** onto the direction of **A**.

line from the tip of **B** onto the direction of **A**, then the line segment B_A is the orthogonal projection of **B** onto line of action of **A**. B_A stands for the scalar projection of **B** onto **A**. From trigonometry, it is obvious from the figure that

$$B_A = B\cos\theta$$

Let $\hat{\mathbf{u}}_A$ be the unit vector in the direction of **A**. Then

$$\mathbf{B}\cdot\hat{\mathbf{u}}_A = \|\mathbf{B}\|\overbrace{\|\hat{\mathbf{u}}_A\|}^{1}\cos\theta = B\cos\theta$$

Comparing this expression with the preceding one leads to the conclusion that

$$B_A = \mathbf{B}\cdot\hat{\mathbf{u}}_A = \mathbf{B}\cdot\frac{\mathbf{A}}{A} \tag{1.12}$$

where $\hat{\mathbf{u}}_A$ is given by Equation 1.1. Likewise, the projection of **A** onto **B** is given by

$$A_B = \mathbf{A}\cdot\frac{\mathbf{B}}{B}$$

Observe that $A_B = B_A$ only if **A** and **B** have the same magnitude.

Example 1.2

Let $\mathbf{A} = \hat{\mathbf{i}} + 6\hat{\mathbf{j}} + 18\hat{\mathbf{k}}$ and $\mathbf{B} = 42\hat{\mathbf{i}} - 69\hat{\mathbf{j}} + 98\hat{\mathbf{k}}$. Calculate

(a) The angle between **A** and **B**;
(b) The projection of **B** in the direction of **A**;
(c) The projection of **A** in the direction of **B**.

Solution

First we make the following individual calculations.

$$\mathbf{A}\cdot\mathbf{B} = (1)(42) + (6)(-69) + (18)(98) = 1392 \tag{a}$$

$$A = \sqrt{(1)^2 + (6)^2 + (18)^2} = 19 \tag{b}$$

$$B = \sqrt{(42)^2 + (-69)^2 + (98)^2} = 127 \tag{c}$$

(a) According to Equation 1.7, the angle between **A** and **B** is

$$\theta = \cos^{-1}\left(\frac{\mathbf{A}\cdot\mathbf{B}}{AB}\right)$$

Substituting (a), (b) and (c) yields

$$\theta = \cos^{-1}\left(\frac{1392}{19\cdot 127}\right) = \boxed{54.77^\circ}$$

(b) From Equation 1.12 we find the projection of **B** onto **A**:

$$B_A = \mathbf{B}\cdot\frac{\mathbf{A}}{A} = \frac{\mathbf{A}\cdot\mathbf{B}}{A}$$

Substituting (a) and (b) we get

$$B_A = \frac{1392}{19} = \boxed{73.26}$$

(c) The projection of **A** onto **B** is

$$A_B = \mathbf{A}\cdot\frac{\mathbf{B}}{B} = \frac{\mathbf{A}\cdot\mathbf{B}}{B}$$

Substituting (a) and (c) we obtain

$$A_B = \frac{1392}{127} = \boxed{10.96}$$

The ***cross product*** of two vectors yields another vector, which is computed as follows,

$$\mathbf{A}\times\mathbf{B} = (AB\sin\theta)\hat{\mathbf{n}}_{AB} \tag{1.13}$$

where θ is the angle between the heads of A and B, and $\hat{\mathbf{n}}_{AB}$ is the unit vector normal to the plane defined by the two vectors. The direction of $\hat{\mathbf{n}}_{AB}$ is determined by the right hand rule. That is, curl the fingers of the right hand from the first vector (**A**) towards the second vector (**B**), and the thumb shows the direction of $\hat{\mathbf{n}}_{AB}$. See Figure 1.7. If we use Equation 1.13 to compute **B** × **A**, then $\hat{\mathbf{n}}_{AB}$ points in the opposite direction, which means

$$\mathbf{B}\times\mathbf{A} = -(\mathbf{A}\times\mathbf{B}) \tag{1.14}$$

Therefore, unlike the dot product, the cross product is not commutative.

The cross product is obtained analytically by resolving the vectors into Cartesian components.

$$\mathbf{A}\times\mathbf{B} = (A_x\hat{\mathbf{i}} + A_y\hat{\mathbf{j}} + A_z\hat{\mathbf{k}})\times(B_x\hat{\mathbf{i}} + B_y\hat{\mathbf{j}} + B_z\hat{\mathbf{k}}) \tag{1.15}$$

Since the set $\hat{\mathbf{i}}\hat{\mathbf{j}}\hat{\mathbf{k}}$ is a mutually perpendicular triad of unit vectors, Equation 1.13 implies that

$$\begin{aligned} &\hat{\mathbf{i}}\times\hat{\mathbf{i}} = \mathbf{0} \quad &\hat{\mathbf{j}}\times\hat{\mathbf{j}} = \mathbf{0} \quad &\hat{\mathbf{k}}\times\hat{\mathbf{k}} = \mathbf{0} \\ &\hat{\mathbf{i}}\times\hat{\mathbf{j}} = \hat{\mathbf{k}} \quad &\hat{\mathbf{j}}\times\hat{\mathbf{k}} = \hat{\mathbf{i}} \quad &\hat{\mathbf{k}}\times\hat{\mathbf{i}} = \hat{\mathbf{j}} \end{aligned} \tag{1.16}$$

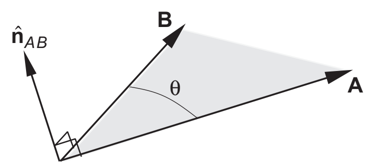

FIGURE 1.7

$\hat{\mathbf{n}}_{AB}$ is normal to both A and B and defines the direction of the cross product **A** × **B**.