Contents

This page intentionally left blank.

PREFACE

This book is designed to support a one-semester course in numerical methods. It has been written for students who want to learn and apply numerical methods in order to solve problems in engineering and science. As such, the methods are motivated by problems rather than by mathematics. That said, sufficient theory is provided so that students come away with insight into the techniques and their shortcomings.

MATLAB© provides a great environment for such a course. Although other environments (e.g., Excel/VBA, Mathcad) or languages (e.g., Fortran 90, C++) could have been chosen, MATLAB presently offers a nice combination of handy programming features with powerful built-in numerical capabilities. On the one hand, its M-file programming environment allows students to implement moderately complicated algorithms in a structured and coherent fashion. On the other hand, its built-in, numerical capabilities empower students to solve more difficult problems without trying to "reinvent the wheel."

The basic content, organization, and pedagogy of the second edition are essentially preserved in the third edition. In particular, the conversational writing style is intentionally maintained in order to make the book easier to read. This book tries to speak directly to the reader and is designed in part to be a tool for self-teaching.

That said, this edition differs from the past edition in three major ways: (1) two new chapters, (2) several new sections, and (3) revised homework problems.

1. **New Chapters.** As shown in 1, I have developed two new chapters for this edition. Their inclusion was primarily motivated by my classroom experience. That is, they are included because they work well in the undergraduate numerical methods course I teach at Tufts. The students in that class typically represent all areas of engineering and range from sophomores to seniors with the majority at the junior level. In addition, we typically draw a few math and science majors. The two new chapters are:

   - **Eigenvalues.** When I first developed this book, I considered that eigenvalues might be deemed an "advanced" topic. I therefore presented the material on this topic at the end of the semester and covered it in the book as an appendix. This sequencing had the ancillary advantage that the subject could be partly motivated by the role of eigenvalues in the solution of linear systems of ODEs. In recent years, I have begun

     to move this material up to what I consider to be its more natural mathematical position at the end of the section on linear algebraic equations. By stressing applications (in particular, the use of eigenvalues to study vibrations), I have found that students respond very positively to the subject in this position. In addition, it allows me to return to the topic in subsequent chapters which serves to enhance the students' appreciation of the topic.

   - **Fourier Analysis.** In past years, if time permitted, I also usually presented a lecture at the end of the semester on Fourier analysis. Over the past two years, I have begun presenting this material at its more natural position just after the topic of linear least squares. I motivate the subject matter by using the linear least-squares approach to fit sinusoids to data. Then, by stressing applications (again vibrations), I have found that the students readily absorb the topic and appreciate its value in engineering and science. It should be noted that both chapters are written in a modular fashion and could be skipped without detriment to the course's pedagogical arc. Therefore, if you choose, you can either omit them from your course or perhaps move them to the end of the semester. In any event, I would not have included them in the current edition if they did not represent an enhancement within my current experience in the classroom. In particular, based on my teaching evaluations, I find that the stronger, more motivated students actually see these topics as highlights. This is particularly true because MATLAB greatly facilitates their application and interpretation.

2. **New Content.** Beyond the new chapters, I have included new and enhanced sections on a number of topics. The primary additions include sections on animation (Chap. 3), Brent's method for root location (Chap. 6), *LU* factorization with pivoting (Chap. 8), *random numbers and Monte Carlo simulation* (Chap. 14), *adaptive quadrature* (Chap. 20),and *event termination of ODEs* (Chap. 23).

3. **New Homework Problems.** Most of the end-of-chapter problems have been modified, and a variety of new problems have been added. In particular, an effort has been made to include several new problems for each chapter that are more challenging and difficult than the problems in the previous edition.

PART ONE Modeling, Computers, and Error Analysis	**PART TWO** Roots and Optimization	**PART THREE** Linear Systems	**PART FOUR** Curve Fitting	**PART FIVE** Integration and Differentiation	**PART SIX** Ordinary Differential Equations
CHAPTER 1 Mathematical Modeling, Numerical Methods, and Problem Solving	CHAPTER 5 Roots: Bracketing Methods	CHAPTER 8 Linear Algebraic Equations and Matrices	CHAPTER 14 Linear Regression	CHAPTER 19 Numerical Integration Formulas	CHAPTER 22 Initial-Value Problems
CHAPTER 2 MATLAB Fundamentals	CHAPTER 6 Roots: Open Methods	CHAPTER 9 Gauss Elimination	CHAPTER 15 General Linear Least-Squares and Nonlinear Regression	CHAPTER 20 Numerical Integration of Functions	CHAPTER 23 Adaptive Methods and Stiff Systems
CHAPTER 3 Programming with MATLAB	CHAPTER 7 Optimization	CHAPTER 10 *LU* Factorization	CHAPTER 16 Fourier Analysis	CHAPTER 21 Numerical Differentiation	CHAPTER 24 Boundary-Value Problems
CHAPTER 4 Roundoff and Truncation Errors		CHAPTER 11 Matrix Inverse and Condition	CHAPTER 17 Polynomial Interpolation		
		CHAPTER 12 Iterative Methods	CHAPTER 18 Splines and Piecewise Interpolation		
		CHAPTER 13 Eigenvalues			

Figure 1: An outline of this edition. The shaded areas represent new material. In addition, several of the original chapters have been supplemented with new topics.

Aside from the new material and problems, the third edition is very similar to the second. In particular, I have endeavored to maintain most of the features contributing to its pedagogical effectiveness including extensive use of worked examples and engineering and scientific applications. As with the previous edition, I have made a concerted effort to make this book as "student-friendly" as possible. Thus, I've tried to keep my explanations straightforward and practical.

Although my primary intent is to empower students by providing them with a sound introduction to numerical problem solving, I have the ancillary objective of making this introduction exciting and pleasurable. I believe that motivated students who enjoy engineering and science, problem solving, mathemati cs—and yes—programming, will ultimately make better professionals. If my book fosters enthusiasm and appreciation for these subjects, I will consider the effort a success.

Acknowledgments. Several members of the McGraw-Hill team have contributed to this project. Special thanks are due to Lorraine Buczek, and Bill Stenquist, and Melissa Leick for their encouragement, support, and direction. Ruma Khurana of MPS Limited, a Macmillan Company also did an outstanding job in the book's final production phase. Last, but not least, Beatrice Sussman once again demonstrated why she is the best copyeditor in the business. During the course of this project, the folks at The MathWorks, Inc., have truly demonstrated their overall excellence as well as their strong commitment to engineering and science education. In particular, Courtney Esposito and Naomi Fernandes of The MathWorks, Inc., Book Program have been especially helpful. The generosity of the Berger family, and in particular Fred Berger, has provided me with the opportunity to work on creative projects such as this book dealing with computing and engineering. In addition, my colleagues in the School of Engineering at Tufts, notably Masoud Sanayei, Lew Edgers, Vince Manno, Luis Dorfmann, Rob White, Linda Abriola, and Laurie Baise, have been very supportive and helpful. Significant suggestions were also given by a number of colleagues. In particular, Dave Clough (University of Colorado–Boulder), and Mike Gustafson (Duke University) provided valuable ideas and suggestions. In addition, a number of reviewers provided useful feedback and advice including Karen Dow Ambtman (University of Alberta), Jalal Behzadi (Shahid Chamran University), Eric Cochran (Iowa State University), Frederic Gibou (University of California at Santa Barbara), Jane Grande-Allen (Rice University), Raphael Haftka (University of Florida), Scott Hendricks (Virginia Tech University), Ming Huang (University of San Diego), Oleg Igoshin (Rice University), David Jack (Baylor University), Clare McCabe (Vanderbilt University), Eckart Meiburg (University of California at Santa Barbara), Luis Ricardez (University of Waterloo), James Rottman (University of California, San Diego), Bingjing Su (University of Cincinnati), Chin-An Tan (Wayne State University), Joseph Tipton (The University of Evansville), Marion W. Vance (Arizona State University), Jonathan Vande Geest (University of Arizona), and Leah J. Walker (Arkansas State University). It should be stressed that although I received useful advice from the aforementioned individuals, I am responsible for any inaccuracies or mistakes you may find in this book. Please contact me via e-mail if you should detect any errors. Finally, I want to thank my family, and in particular my wife, Cynthia, for the love, patience, and support they have provided through the time I've spent on this project.

Steven C. Chapra
Tufts University
Medford, Massachusetts
steven.chapra@tufts.edu

PEDAGOGICAL TOOLS

Theory Presented as It Informs Key Concepts. The text is intended for Numerical Methods users, not developers. Therefore, theory is not included for "theory's sake," for example no proofs. Theory is included as it informs key concepts such as the Taylor series, convergence, condition, etc. Hence, the student is shown how the theory connects with practical issues in problem solving.

Introductory MATLAB Material. The text includes two introductory chapters on how to use MATLAB. Chapter 2 shows students how to perform computations and create graphs in MATLAB's standard command mode. Chapter 3 provides a primer on developing numerical programs via MATLAB M-file functions. Thus, the text provides students with the means to develop their own numerical algorithms as well as to tap into MATLAB's powerful built-in routines.

Algorithms Presented Using MATLAB M-files. Instead of using pseudocode, this book presents algorithms as well-structured MATLAB M-files. Aside from being useful computer programs, these provide students with models for their own M-files that they will develop as homework exercises.

Worked Examples and Case Studies. Extensive worked examples are laid out in detail so that students can clearly follow the steps in each numerical computation. The case studies consist of engineering and science applications which are more complex and richer than the worked examples. They are placed at the ends of selected chapters with the intention of (1) illustrating the nuances of the methods, and (2) showing more realistically how the methods along with MATLAB are applied for problem solving.

Problem Sets. The text includes a wide variety of problems. Many are drawn from engineering and scientific disciplines. Others are used to illustrate numerical techniques and theoretical concepts. Problems include those that can be solved with a pocket calculator as well as others that require computer solution with MATLAB.

Useful Appendices and Indexes. Appendix A contains MATLAB commands, and Appendix B contains M-file functions.

Textbook Website. A text-specific website is available at www.mhhe.com/chapra. Resources include the text images in PowerPoint, M-files, and additional MATLAB resources. This page intentionally left blank.

This page intentionally left blank.

Modeling, Computers,and Error Analysis

0.1. MOTIVATION

What are numerical methods and why should you study them?

Numerical methods are techniques by which mathematical problems are formulated so that they can be solved with arithmetic and logical operations. Because digital computers excel at performing such operations, numerical methods are sometimes referred to as computer mathematics.

In the pre–computer era, the time and drudgery of implementing such calculations seriously limited their practical use. However, with the advent of fast, inexpensive digital computers, the role of numerical methods in engineering and scientific problem solving has exploded. Because they figure so prominently in much of our work, I believe that numerical methods should be a part of every engineer's and scientist's basic education. Just as we all must have solid foundations in the other areas of mathematics and science, we should also have a fundamental understanding of numerical methods. In particular, we should have a solid appreciation of both their capabilities and their limitations. Beyond contributing to your overall education, there are several additional reasons why you should study numerical methods:

1. Numerical methods greatly expand the types of problems you can address. They are capable of handling large systems of equations, nonlinearities, and complicated geometries that are not uncommon in engineering and science and that are often impossible to solve analytically with standard calculus. As such, they greatly enhance your problem-solving skills.

2. Numerical methods allow you to use "canned" software with insight. During your career, you will invariably have occasion to use commercially available prepackaged computer programs that involve numerical methods. The intelligent use of these programs is greatly enhanced by an understanding of the basic theory underlying the methods. In the absence of such understanding, you will be left to treat such packages as "black boxes" with little critical insight into their inner workings or the validity of the results they produce.

3. Many problems cannot be approached using canned programs. If you are conversant with numerical methods, and are adept at computer programming, you can design your own programs to solve problems without having to buy or commission expensive software.

4. Numerical methods are an efficient vehicle for learning to use computers. Because numerical methods are expressly designed for computer implementation, they are ideal for illustrating the computer's powers and limitations. When you successfully implement numerical methods on a computer, and then apply them to solve otherwise intractable problems, you will be provided with a dramatic demonstration of how computers can serve your professional development. At the same time, you will also learn to acknowledge and control the errors of approximation that are part and parcel of large-scale numerical calculations.

5. Numerical methods provide a vehicle for you to reinforce your understanding of mathematics. Because one function of numerical methods is to reduce higher mathematics to basic arithmetic operations, they get at the "nuts and bolts" of some otherwise obscure topics. Enhanced understanding and insight can result from this alternative perspective.

With these reasons as motivation, we can now set out to understand how numerical methods and digital computers work in tandem to generate reliable solutions to mathematical problems. The remainder of this book is devoted to this task.

0.2. PART ORGANIZATION

This book is divided into six parts. The latter five parts focus on the major areas of numerical methods. Although it might be tempting to jump right into this material, *Part One* consists of four chapters dealing with essential background material.

Chapter 1 provides a concrete example of how a numerical method can be employed to solve a real problem. To do this, we develop a *mathematical model* of a free-falling bungee jumper. The model, which is based on Newton's second law, results in an ordinary differential equation. After first using calculus to develop a closed-form solution, we then show how a comparable solution can be generated with a simple numerical method. We end the chapter with an overview of the major areas of numerical methods that we cover in Parts Two through Six.

Chapters 2 and 3 provide an introduction to the MATLAB© software environment. *Chapter 2* deals with the standard way of operating MATLAB by entering commands one at a time in the so-called calculator, or command, mode. This interactive mode provides a straightforward means to orient you to the environment and illustrates how it is used for common operations such as performing calculations and creating plots.

Chapter 3 shows how MATLAB's programming mode provides a vehicle for assembling individual commands into algorithms. Thus, our intent is to illustrate how MATLAB serves as a convenient programming environment to develop your own software.

Chapter 4 deals with the important topic of error analysis, which must be understood for the effective use of numerical methods. The first part of the chapter focuses on the *roundoff errors* that result because digital computers cannot represent some quantities exactly. The latter part addresses *truncation errors* that arise from using an approximation in place of an exact mathematical procedure.

Chapter 1

Mathematical Modeling, Numerical Methods, and Problem Solving

CHAPTER OBJECTIVES

The primary objective of this chapter is to provide you with a concrete idea of what numerical methods are and how they relate to engineering and scientific problem solving. Specific objectives and topics covered are

- Learning how mathematical models can be formulated on the basis of scientific principles to simulate the behavior of a simple physical system.
- Understanding how numerical methods afford a means to generate solutions in a manner that can be implemented on a digital computer.
- Understanding the different types of conservation laws that lie beneath the models used in the various engineering disciplines and appreciating the difference between steady-state and dynamic solutions of these models.
- Learning about the different types of numerical methods we will cover in this book.

YOU'VE GOT A PROBLEM

Suppose that a bungee-jumping company hires you. You're given the task of predicting the velocity of a jumper
1.1 as a function of time during the free-fall part of the jump. This information will be used as part of a larger analysis to determine the length and required strength of the bungee cord for jumpers of different mass. You know from your studies of physics that the acceleration should be equal to the ratio of the force to the mass (Newton's second law). Based on this insight and your knowledge of physics and fluid mechanics, you develop the following mathematical model for the rate of change of velocity with respect to time,

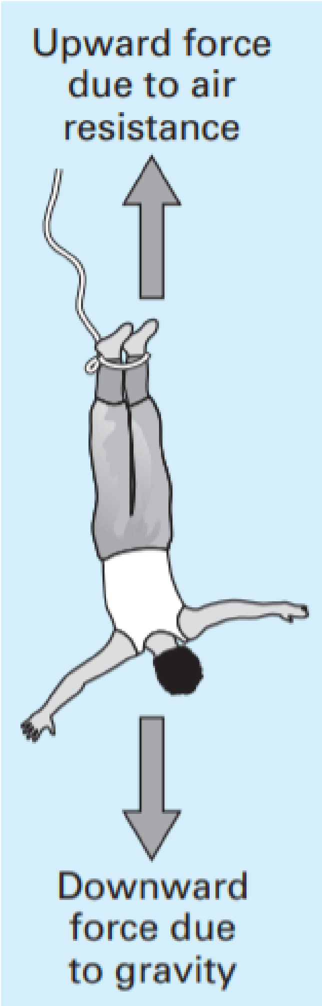

Figure 1.1: Forces acting on a free-falling bungee jumper

$\frac{dv}{dt} = g - \frac{c_d}{m}v^2$ where $v =$ downward vertical velocity (m/s), $t =$ time (s), $g =$ the acceleration due to gravity ($\cong$ 9.81 m/s2), $c_d = a$ lumped drag coefficient (kg/m), and $m =$ the jumper's mass (kg). The drag coefficient is called "lumped" because its magnitude depends on factors such as the jumper's area and the fluid density (see Sec 1.4).

Because this is a differential equation, you know that calculus might be used to obtain an analytical or exact solution for v as a function of t. However, in the following pages, we will illustrate an alternative solution approach. This will involve developing a computeroriented numerical or approximate solution.

Aside from showing you how the computer can be used to solve this particular problem, our more general objective will be to illustrate (a) what numerical methods are and (b) how they figure in engineering and scientific problem solving. In so doing, we will also show how mathematical models figure prominently in the way engineers and scientists use numerical methods in their work.

1.1. A SIMPLE MATHEMATICAL MODEL