Preface

About the Text

We developed this text for a sequence of courses on the theory and application of numerical approximation techniques. We designed it primarily for junior-level mathematics, science, and engineering majors who have completed at least the first year of the standard college calculus sequence. Familiarity with the fundamentals of matrix algebra and differential equations is useful, but we present introductory material on these topics so that such courses need not be prerequisites.

Previous editions of *Numerical Analysis* have been used in a wide variety of situations. In some cases, the mathematical analysis underlying the development of approximation techniques was emphasized rather than the methods themselves; in others, the emphasis was reversed. The book has also been used as a core reference for beginning graduate level courses in engineering and computer science programs, and in first-year courses in introductory analysis offered at international universities. We have adapted the book to fit these diverse users without compromising our original purpose:

> *To introduce modern approximation techniques; to explain how, why, and when they can be expected to work; and to provide a foundation for further study of numerical analysis and scientific computing.*

The book contains sufficient material for a full year of study, but we expect many instructors will use the text only for a single-term course. In such a single-term course, students learn to identify the types of problems that require numerical techniques for their solution and see examples of the error propagation that can occur when numerical methods are applied. They accurately approximate the solution of problems that cannot be solved exactly and learn typical techniques for estimating error bounds for the approximations. The remainder of the text then serves as a reference for methods that are not considered in the course. Either the full-year or single-course treatment is consistent with the aims of the text.

Virtually every concept in the text is illustrated by example, and this edition contains nearly 2500 class-tested exercises ranging from elementary applications of methods and algorithms to generalizations and extensions of their theory. In addition, the exercise sets include numerous applied problems from diverse areas of engineering, as well as from the physical, computer, biological, and social sciences. The applications chosen clearly and concisely demonstrate how numerical techniques can be, and often must be, applied in real-life situations.

A number of software packages, Computer Algebra Systems (CAS), have been developed to produce symbolic mathematical computations. Maple© and Mathematica© are predominant among these in the academic environment. Student versions of these software packages are available at reasonable prices for most common computer systems. Although there are significant differences among the packages, both in performance and price, all can perform standard algebra and calculus operations. Having a symbolic computation package available can be very useful in the study of approximation techniques, since exact solutions can often be obtained quite easily using symbolic computation.

The results in most of our examples and exercises have been generated using problems for which exact values *can* be determined, since this permits the performance of the approximation method to be monitored. In addition, for many numerical techniques the error analysis requires bounding a higher ordinary or partial derivative of a function, which can be a tedious task and one that is not particularly instructive once the techniques of calculus have been mastered. Derivatives can be quickly obtained symbolically, and a little insight often permits a symbolic computation to aid in the bounding process as well.

We have chosen Maple as our standard package because of its wide distribution and the helpful technical support they have provided, but Mathematica can be substituted with only minor modifications. Examples and exercises have been added whenever we felt that a CAS would be of significant benefit, and we have discussed the approximation methods that Maple employs when it is unable to solve a problem exactly.

New for This Edition

It has been more than 25 years since the first edition of this book was published, which came out in the decade after major advances in numerical techniques were made to reflect the new widespread availability of computer equipment. In each of our revisions of the book we have added new techniques in an attempt to keep our treatment current. This revision has added no new sections; instead, we have fine-tuned the presentation of the material to make the book both more useful to the instructor and more interesting to the reader. For example, we have added nearly 200 exercises, most of these of a relatively routine nature, to give instructors a better choice for their problem assignments. We have also increased the page size of the book slightly which permitted us to add more than 150 historical and instructional comments to the margins.

In our first edition we introduced a feature that was innovative and somewhat controversial. Instead of presenting our approximation techniques in a specific programming language (FORTRAN was dominant at the time), we gave algorithms in pseudo code that would lead to a well-structured program in a variety of languages. Beginning with the second edition we listed programs in specific languages in the *Instructors Manual* for the book, and the number of these languages increased in subsequent editions. We now have the programs coded and available online in most common programming languages and CAS work sheets. For this addition we have added Java implementation for the algorithms, and all of these are on the web site for the book:

http://www.as.ysu.edu/~faires/Numerical-Analysis/

On this web site we will place updated programs as the software changes and post responses to comments made by users of the book. We will also add new material that might be included in subsequent editions in PDF format that users can download. Our hope is that this will extend the life of the eighth edition while keeping the material current.

Algorithms

As in previous editions, we give a detailed, structured algorithm for each method in the text without program listing. The algorithms are in a form that can be coded, even by those with limited programming experience.

In this edition we have chosen to put the programs on the web site for the book, since our reviewers suggested that this was preferable to having a disk included with the book. For each algorithm there is a program written in Fortran, Pascal, C, and Java. In addition, we have coded the programs using Maple and Mathematica, as well as in MATLAB©, a computer software package that is widely used for linear algebra applications. This should ensure that a set of programs is available for most common computing systems.

A *Student Study Guide* is available with this edition that contains worked-out solutions to many of the problems. Please contact the authors if you are interested in the *Guide*.

Brooks/Cole can provide instructors with an *Instructor's Manual* that provides answers and solutions to all the exercises in the book. Computation results in the *Instructor's Manual* were regenerated for this edition, using the programs on the disk to ensure compatibility among the various programming systems.

The algorithms in the text lead to programs that give correct results for the examples and exercises in the text, but no attempt was made to write general-purpose professional software. Specifically, the algorithms are not always written in a form that leads to the *most efficient* program in terms of either time or storage requirements. When a conflict occurred between writing an extremely efficient algorithm and writing a slightly different one that better illustrates the important features of the method, the latter path was invariably taken.

About the Programs

For each algorithm there is a C, Fortran, Java, Maple, Mathematica, MATLAB, and Pascal program, and for some of these systems there are multiple programs that depend on the particular version of the software being run. Every program is illustrated with a sample problem that is closely correlated to the text. This permits the program to be run initially in the language of your choice to see the form of the input and output. The programs can then be modified for other problems by making minor changes. The form of the input and output are, as nearly as possible, the same in each of the programming systems. This permits an instructor using the programs to discuss them generically, without regard to the particular programming system an individual student uses.

The programs are designed to run on a minimally configured computer. For most of the programming systems you will need appropriate software, such as a compiler for Pascal, Fortran, and C, or one of the computer algebra systems (Maple, Mathematica, and MATLAB). The Java implementations are somewhat of an exception. You need the system to run the programs, but Java can be freely downloaded from various sites. The best way to obtain Java is to use a search engine to search on the name, choose a download site, and follow the instructions for that site.

All of the programs are given as ASCII files or worksheets. They can be altered using any editor or word processor that creates a standard ASCII file. (These are also commonly called "Text Only" files.) Extensive README files are included with the program files so that the peculiarities of the various programming systems can be individually addressed. The README files are presented both in ASCII format and as PDF files. As new software is developed, the algorithms will be updated and placed on the web site for the book.

Suggested Course Outlines

Numerical Analysis is designed to allow instructors flexibility in the choice of topics, as well as in the level of theoretical rigor and in the emphasis on applications. In line with these aims, we provide detailed references for the results that are not demonstrated in the text and for the applications that are used to indicate the practical importance of the methods. The text references cited are those most likely to be available in college libraries and have been updated to reflect recent editions. We also include quotations from original research papers when we feel this material is accessible to our intended audience.

The following flowchart indicates chapter prerequisites. The only deviation from this chart is described in the footnote at the bottom of the first page of Section 3.4. Most of the possible sequences that can be generated from this chart have been taught by the authors at Youngstown State University.