

Preface

This book is designed for students and researchers who seek practical knowledge of modern techniques in scientific computing. The text aims to provide an in-depth treatment of fundamental issues and methods, and the reasons behind success and failure of numerical software. On one hand, we avoid an extensive, encyclopedic, heavily theoretical exposition, and try to get to current methods, issues, and software fairly quickly. On the other hand, this is by no means a quick recipe book, since we feel that the introduction of algorithms requires adequate theoretical foundation: having a solid basis enables those who need to apply the techniques to successfully design their own solution approach for any nonstandard problems they may encounter in their work.

There are many books on scientific computing and numerical analysis, and a natural question here would be why we think that yet another text is necessary. It is mainly because we feel that in an age where yesterday's concepts are not necessarily today's truths, scientific computing is constantly redefining itself and is now positioned as a discipline truly at the junction of mathematics, computer science, and engineering. Books that rely heavily on theory, or on algorithm recipes, do not quite capture the current state of this broad and dynamic area of research and application. We thus take an algorithmic approach and focus on techniques of a high level of applicability to engineering, computer science, and industrial mathematics practitioners. At the same time, we provide mathematical justification throughout for the methods introduced. While we refrain from a theorem–proof type of exposition, we construct the theory behind the methods in a systematic and fairly complete fashion. We make a strong effort to emphasize computational aspects of the algorithms discussed by way of discussing their efficiency and their computational limitations.

This book has been developed from notes for two courses taught by the Department of Computer Science at the University of British Columbia (UBC) for over 25 years. The first author developed a set of notes back in the late 1970s. These were based on the books by Conte and de Boor [13], Dahlquist and Björck [15], and others, as well as on notes by Jim Varah. Improvements to these notes were subsequently made by Ian Cavers, by the second author, and by Dhavide Aruliah. A substantial addition of material was made by the two authors in the last few years. A few of our colleagues (Jack Snoeyink, Oliver Dorn, Ian Mitchell, Kees van den Doel and Tyrone Rees at UBC, Jim Lambers at Stanford University, Marsha Berger and Michael Overton at NYU, Edmond Chow at Columbia University, André Weidemann at Stellenbosch, and others) have used the notes and provided us with invaluable feedback.

Most of the material contained in this book can be covered in two reasonably paced semesterial courses. A possible model here is the one that has been adopted at UBC: one course covers Chapters 3 to 9, while another concentrates on Chapters 10 to 16. The two parts are designed so that they do not heavily rely on each other, although we have found it useful to include the material of Chapter 1 and a subset of Chapter 2 in both semesterial courses. More advanced material, contained in sections denoted by an asterisk, is included in the text as a natural extension. However, with one or two exceptions, we have not taught the advanced material in the two basic courses.

Another use of this text is for a breadth-type introductory course on numerical methods at the graduate level. The target audience would be beginning graduate students in a variety of disci-

plines including computer science, applied mathematics, and engineering, who start graduate school without having taken undergraduate courses in this area. Such a course would also use our more advanced sections and possibly additional material.

The division of the material into chapters can be justified by the type of computational errors encountered. These concepts are made clear in the text. We start off with *roundoff error*, which appears in all the numerical processes considered in this book. Chapters 1 and 2 systematically cover it, and later chapters, such as 5 and 6, contain assessments of its effect. Chapter 3 and Chapters 7 to 9 are concerned mainly with iterative processes, and as such, the focus is on the iteration, or *convergence error*. Finally, control and assessment of *discretization error* is the main focus in Chapters 10 to 16, although the other error types do arise there as well.

We illustrate our algorithms using the programming environment of MATLAB and expect the reader to become gradually proficient in this environment while (if not before) learning the material covered in this book. But at the same time, we use the MATLAB language and programming environment merely as a tool, and we refrain from turning the book into a language tutorial.

Each chapter contains exercises, ordered by section. In addition, there is an exercise numbered 0 which consists of several review questions intended for self-testing and for punctuating the process of reading and absorbing the presented material. Instructors should resist the temptation to answer these review questions for the students.

We have made an effort to make the chapters fairly independent of one another. As a result, we believe that readers who are interested in specific topics may often be able to settle for reading almost exclusively only the chapters that are relevant to the material they wish to pursue.

We are maintaining a project webpage for this book which can be found at

<http://www.siam.org/books/cs07>

It contains links to our programs, solutions to selected exercises, errata, and more.

Last but not at all least, many generations of students at UBC have helped us over the years to shape, define, and debug this text. The list of individuals who have provided crucial input is simply too long to mention. This book is dedicated to you, our students.