Preface

This book is intended for sophomore-level engineering and science students interested in numerical methods for solving problems such as differential equations, eigenvalue problems, linear systems, etc. and could be appropriate for mathematics majors as well if the instructor supplements this text with more proofs. It has evolved from our lecture notes over the many times we taught the course since 2010.

We assume a knowledge of calculus through multivariable calculus (typically called Calculus III in the United States), and we teach the course at Clemson with a first course in ordinary differential equations being a corequisite. Also, we assume that students have at least seen and used MATLAB in some minimal capacity.

We decided to write this book because we wanted the text for this course to have the following:

1. Low cost. Many books that cover similar material cost upward of $100 or even $150, and we refuse to be part of that. Hence we have chosen a publisher that provides a much lower cost.
2. Simple programming examples. This is an introductory programming course, and students need to learn the basics before all the bells and whistles. Hence whenever possible, we give code that is as simple as possible, without overcomplicating the main ideas and without excessive commenting.
3. Cover the fundamentals. For any given problem for which we use numerical methods, there are often dozens of approaches. We believe an introductory text should focus on the basics, and so we give just the most common approaches and discuss them in more detail.

Second edition

In the second edition of this book, we made the following additions/changes on top of minor corrections:

- Various new and improved figures
- Many more exercises
- Fixes to proofs in Chapters 6 and 9
- New content in Chapter 5 (inverse interpolation, fixed point theory, Anderson acceleration) and Chapter 6 (1 norm)

We would greatly appreciate any users of this book to point out typos or other mistakes to us. We would also appreciate any other constructive criticism regarding the presentation of the material.

Software

Algorithms given in the text are written in the language of MATLAB and Octave. Currently at Clemson, all students have free access to MATLAB. Octave is a free version of

https://doi.org/10.1515/9783110988451-201

MATLAB, which has almost all of the same functionality. Newer versions of MATLAB have more bells and whistles, but for the purposes of this book, either MATLAB or Octave can be used.

We have created a website for the codes used in this book, where all MATLAB/Octave codes from the text can be downloaded. We have also posted Python versions of the codes: http://www.math.clemson.edu/~heister/scicompbook/

About the title image

The title image of the second edition is a rendering of a three-dimensional finite element simulation of convection in the Earth's mantle. The simulation covers one billions years of plate motion history that is prescribed at the surface, and the graphic shown is the final state. Plumes are shown in red-yellow (using an isosurface of high temperature), and subducted slabs are shown in blue (using an isosurface of low temperature).

The simulation is done using the ASPECT code available at https://aspect.geodynamics.org by Rene Gassmöller and Juliane Dannberg (University of Florida). The computation uses various numerical methods: The time-dependent partial differential equations are discretized in time similar to backward Euler in Chapter 10. Each time-step involves solving a nonlinear system using Newton's method (Chapter 5). The linear systems are created using the finite element method, which uses Gauss quadrature (Chapter 9) and techniques similar to Chapter 4. Each linear system is solved using an iterative method in parallel. Direct solvers like Gauss elimination are not feasible because the linear system has about 125 million rows/columns.