CMS/CAIMS Books in Mathematics

Canadian
Mathematical Society
Société mathématique
du Canada

David E. Stewart

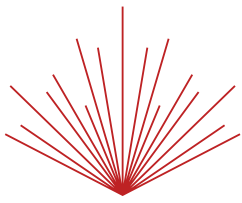
Numerical Analysis: A Graduate Course

CAIMS
SCMAI

CMS/CAIMS Books in Mathematics

Volume 4

CMS/CAIMS Books in Mathematics is a collection of monographs and graduate-level textbooks published in cooperation jointly with the Canadian Mathematical Society- Societé mathématique du Canada and the Canadian Applied and Industrial Mathematics Society-Societé Canadienne de Mathématiques Appliquées et Industrielles. This series offers authors the joint advantage of publishing with two major mathematical societies and with a leading academic publishing company. The series is edited by Karl Dilcher, Frithjof Lutscher, Nilima Nigam, and Keith Taylor. The series publishes high-impact works across the breadth of mathematics and its applications. Books in this series will appeal to all mathematicians, students and established researchers. The series replaces the CMS Books in Mathematics series that successfully published over 45 volumes in 20 years.



David E. Stewart

Numerical Analysis: A Graduate Course

Springer

David E. Stewart
Department of Mathematics
University of Iowa
Iowa, IA, USA

ISSN 2730-650X ISSN 2730-6518 (electronic)
CMS/CAIMS Books in Mathematics
ISBN 978-3-031-08120-0 ISBN 978-3-031-08121-7 (eBook)
https://doi.org/10.1007/978-3-031-08121-7

Mathematics Subject Classification: 65-01

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Numerical analysis has advanced greatly since it began as a way of creating methods to approximate answers to mathematical questions. This book aims to bring students closer to the frontier regarding the numerical methods that are used. But this book is not only about newer, as well as "classical", numerical methods. Rather the aim is to also explain how and why they work, or fail to work. This means that there is a significant amount of theory to be understood. Simple analyses can result in methods that usually work, but can then fail in certain circumstances, sometimes catastrophically. The causes of success of a numerical algorithm and its failure are both important. Without understanding the underlying theory, the reasons for a method's success and failure remain mysterious, and we do not have a means to determine how to fix the problem(s).

In this way, numerical analysis is a dialectic[1] between practice and theory; the practice being computation and programming, and the theory being mathematical analysis based on our model of computation. While we do indeed prove theorems in numerical analysis, the assumptions made in these theorems may not hold in many situations. Also, the conclusions may involve statements of the form "as n goes to infinity" (or "as h goes to zero") while in actual computations n might not be especially large (or h especially small). Numerical analysis will sometimes ignore errors that we know exist (like roundoff error in the analysis of a method for solving differential equations). This is usually based on an understanding that some sources of errors are insignificant in a particular situation. Of course, there will be situations where roundoff error should be considered in the solution of differential equations, but only if the step size becomes unusually small. In that case, new analyses, and even new methods, may be necessary.

[1] *Dialectic* is a dialog between a claim (a *thesis*) and counter-claims (an *antithesis*) hopefully leading to a new understanding (a *synthesis*) that incorporates both the original claim and the counter-claims. The synthesis is expected to give further understanding, but will itself eventually meet counter-claims.

Inspiration for this book has been found in the books of Atkinson and Han [11], Atkinson [13], Sauer [228], and Stoer and Bulirsch [241]. However, we wish to include current issues and interests that were not addressed in these books.

We aim to present numerical methods and their analysis in the context of modern applications and models. For example, the standard asymptotic error analysis of differential equations gives no advantage to implicit methods, which have a much larger computational cost. But for "stiff" problems there is a clear, and often decisive, advantage to implicit methods. While "stiffness" can be hard to quantify, it is also common in applications. We also wish to emphasize multivariate problems alongside single-variable problems: multivariate problems are crucial for partial differential equations, optimization, and integration over high-dimensional spaces. We deal with issues regarding randomness, including pseudo-random number generators, stochastic differential equations, and randomized algorithms. Stochastic differential equations meet a need for incorporating randomness into differential equations. High-dimensional integration is needed for studying questions and models in data science and simulation.

To summarize, I believe numerical analysis must be understood and taught in the context of applications, not simply as a discipline devoted solely to its own internal issues. Rather, these internal issues arise from understanding the common ground between analysis and applications. This is where the future of the discipline lies.

I would like to thank the many people who have been supportive of this effort, or contributed to it in some way. I would like to thank (in alphabetical order) Jeongho Ahn, Kendall Atkinson, Bruce Ayati, Ibrahim Emirahmetoglu, Koung-Hee Leem, Paul Muhly, Ricardo Rosado-Ortiz, and Xueyu Zhu. My wife, Suely Oliveira, has a special thanks for both encouraging this project and having the patience for me to see it through. Finally, I would like to thank the staff at Springer for their interest and support for this book, most especially Donna Chernyk.

How to Use This Book:

Numerical analysis is a combination of theory and practice. The theory is a mixture of calculus and analysis with some algorithm analysis thrown in. Practice is computation and programming. The algorithms in the book are shown as pseudo-code. Working code for MATLAB and/or Julia can be found at https://github.com/destewart2022/NumerAnal-Gradbook. The exercises are intended to develop both, and students need practice at both.

Like most intellectual disciplines, numerical analysis is more a spiral than a straight line. There is no linear ordering of the topics that makes complete sense. Thus teaching from this book should not be a matter of starting at one cover and ending at the other. In any case, there is probably too much material and an instructor must of necessity choose what they wish to emphasize. Matrix

computations are foundational, but even just focusing on this could easily take a semester or more. Differential equations arise in many, many applications, but the technical issues in partial differential equations can be daunting. The treatment here aims to be accessible without "dumbing down" the material. Students in data science may want to focus on optimization, high-dimensional approximation, and high-dimensional integration. Randomness finds its way into many applications, whether we wish it or not. So, here is a plan that you might consider when you first teach from this book:

- Chapter 1: A little on computing machinery to get started, floating point arithmetic, norms for vectors and matrices, and at least one-variable Taylor series with remainder.
- Chapter 2: LU factorization for linear systems, linear least squares via the normal equations and the QR factorization as a black box. Eigenvalues can wait until later.
- Chapter 3: Bisection, fixed-point, Newton, and secant methods are the foundation; guarded multivariate Newton and one-variable hybrid methods give good examples of how to modify algorithms for better reliability.
- Chapter 4: Polynomial interpolation is so central that you need to cover this well, including the error formula and the Runge phenomenon; the Weierstrass and Jackson theorems (without proof) give a sense of rate of convergence. Cubic splines give useful alternatives to plain polynomials. Lebesgue numbers may appear abstract, but give a good sense of the reliability of interpolation schemes. Radial basis functions give an entry into high-dimensional approximation.
- Chapter 5: Simple ideas can go a long way, but "integrate the interpolant" is a central idea; multivariate integration is also valuable here if you want to use it for partial differential equations.
- Chapter 6: Basic methods for solving ordinary differential equations are still very useful, although the revolution brought about but John Butcher's approach to Runge–Kutta methods is worth a look—if you have time. Partial differential equations need some more set-up time, but are worthwhile for more advanced students, or a second time around. The scale of the problems for partial differential equations means that you should point your students back to Chapter 2 on how to solve large linear systems.
- Chapter 7: Randomness is important, and some statistical computation using SVDs may be a doorway to these issues. Random algorithms are also very important, but often involve advanced ideas.
- Chapter 8: Optimization has become more important in a number of areas, and including it is an option to consider. If your students want to do machine learning, some outline of the algorithms is available here.

A second course could be focused on specific issues. A machine learning focus could include iterative methods and SVDs for matrix computations in Chapter 2, radial basis functions from Chapter 4, high-dimensional integration from Chapter 5, methods for large-scale optimization from Chapter 8, rounded out with some

analysis of random algorithms from Chapter 7. A simulation-based course would focus on approximation and interpolation in two or three dimensions from Chapter 4, multi-dimensional integration from Chapter 5, much of the material from Chapter 6 on differential equations, both ordinary and partial. Uncertainty quantification could be served by starting with Chapter 7, progressing to iterative methods for large linear systems in Chapter 2, radial basis functions in Chapter 4, perhaps some partial differential equations in Chapter 6 and optimization in Chapter 8.

Notes on Notation:

Scalars are usually denoted by lower case italic letters (such as x, y, z, α, β) while vectors are usually denoted by lower case bold letters (such as $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$). Matrices are usually denoted by upper case italic letters (X, Y, A, B). Entries of a vector $\boldsymbol{x}$ are scalars, and so denoted x_i; entries of a matrix A are denoted a_{ij}. Matrices and vectors usually have indexes that are integers $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ for an $m \times n$ matrix. Sometimes it is convenient to have index sets that are different, so a matrix $A = [a_{ij} | i \in R, j \in C]$ can have row index set R and column index set C, and $\boldsymbol{x} = [x_i | i \in C]$ has index set C. This means $A\boldsymbol{x} = [\sum_{j \in C} a_{ij} x_j | i \in R]$ is the matrix–vector product. Just as A^T is used to denote the transpose of A, A^{-T} is the inverse of A^T, which is also the transpose of A^{-1}.

Functions are usually introduced by naming them. For example, the squaring function can be introduced as $q(x) = x^2$. Functions of several variables can be introduced as $f(x, y) = x^2 + y^2$ or using vectors as $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x}$. Anonymous functions can be introduced as $\boldsymbol{x} \mapsto \boldsymbol{x}^T\boldsymbol{x}$.

Pseudo-code uses "$\leftarrow$" to assign a value to a variable (such as $x \leftarrow y$ assigns the value of y to x) while "=" tests for equality (where $x = y$ returns *true* if x and y have the same value).

In some occasions, ":=" is used to define a quantity of function where using "=" might be ambiguous. The sets $\mathbb{R}$, $\mathbb{C}$, and $\mathbb{Z}$ are understood to be the set of real numbers, the set of complex numbers, and the set of integers, respectively.

Iowa, USA David E. Stewart

Contents