Numerical Analysis and Scientific Computing Series

NUMERICAL METHODS FOR SCIENTISTS AND ENGINEERS

With Pseudocodes

Zekeriya Altaç

A Chapman & Hall Book

CRC Press
Taylor & Francis Group

Numerical Methods for Scientists and Engineers

Numerical Methods for Scientists and Engineers: With Pseudocodes is designed as a primary textbook for a one-semester course on Numerical Methods for sophomore or junior-level students. It covers the fundamental numerical methods required for scientists and engineers, as well as some advanced topics which are left to the discretion of instructors.

The objective of the text is to provide readers with a strong theoretical background on numerical methods encountered in science and engineering, and to explain how to apply these methods to practical, real-world problems. Readers will also learn how to convert numerical algorithms into running computer codes.

Features

- Numerous pedagogic features including exercises, "pros and cons" boxes for each method discussed, and rigorous highlighting of key topics and ideas
- Suitable as a primary text for undergraduate courses in numerical methods, but also as a reference to working engineers
- A Pseudocode approach that makes the book accessible to those with different (or no) coding backgrounds, which does not tie instructors to one particular language over another
- A dedicated website featuring additional code examples, quizzes, exercises, discussions, and more: https://github.com/zaltac/NumMethodsWPseudoCodes
- A complete Solution Manual and PowerPoint Presentations are available (free of charge) to instructors at www.routledge.com/9781032754741.

Zekeriya Altaç is currently a professor of mechanical engineering at Eskişehir Osmangazi University and he received his B.Sc. degree in 1983 from Istanbul Technical University in Turkey. He received a scholarship from the Ministry of Education of Turkey to study nuclear engineering in the USA. He attended Iowa State University and earned his M.Sc. (1986) and Ph.D. (1989) in Nuclear Engineering. Upon returning to Turkey, he worked as an assistant professor at the Mechanical Engineering Department of Anadolu University between 1989 and 1992 in Eskişehir, Turkey. Later, he joined the Department of Mechanical Engineering at Eskişehir Osmangazi University as an associate professor in 1993 and received a tenure-track position in 1998, where he is currently working. His areas of interest and expertise are computational nuclear reactor physics, computational fluid mechanics and heat transfer, and numerical (computational) methods in general. He is the author of over 100 journal articles, conference and research papers, books, and book translations. The author has books in Turkish on computer programming (BASIC, FORTRAN) and the application of CFD software (ANSYS, FLUENT). Most of his research deals with developing efficient numerical methods and/or employing computational techniques in practical scientific and engineering problems. He served as an advisor at the Von Karman Institute Technical Advisory Committee (VKI-TAC) in Brussels, Belgium, between 2009 and 2011. He has also served as Eskişehir Osmangazi University's vice president (2007-2011), as head of the Mechanical Engineering Department (2012-2021), and held positions on numerous conferences, symposiums, and academic committees, such as the engineering college board and university senate.

Numerical Analysis and Scientific Computing Series
Series Editors:
Frederic Magoules, Choi-Hong Lai

About the Series
This series, comprising of a diverse collection of textbooks, references, and handbooks, brings together a wide range of topics across numerical analysis and scientific computing. The books contained in this series will appeal to an academic audience, both in mathematics and computer science, and naturally find applications in engineering and the physical sciences.

Modelling with Ordinary Differential Equations
A Comprehensive Approach
Alfio Borzì

Numerical Methods for Unsteady Compressible Flow Problems
Philipp Birken

A Gentle Introduction to Scientific Computing
Dan Stanescu, Long Lee

Introduction to Computational Engineering with MATLAB
Timothy Bower

An Introduction to Numerical Methods
A MATLAB® Approach, Fifth Edition
Abdelwahab Kharab, Ronald Guenther

The Sequential Quadratic Hamiltonian Method
Solving Optimal Control Problems
Alfio Borzì

Advances in Theoretical and Computational Fluid Mechanics
Existence, Blow-up, and Discrete Exterior Calculus Algorithms
Terry Moschandreou, Keith Afas, Khoa Nguyen

Stochastic Methods in Scientific Computing
From Foundations to Advanced Techniques
Massimo D'Elia, Kurt Langfeld, Biagio Lucini

For more information about this series please visit: https://www.crcpress.com/Chapman--HallCRC-Numerical-Analysis-and-Scientific-Computing-Series/book-series/CHNUANSCCOM

Numerical Methods for Scientists and Engineers

With Pseudocodes

Zekeriya Altaç

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

Designed cover image: ShutterStock Images

First edition published 2025
by CRC Press
2385 NW Executive Center Drive, Suite 320, Boca Raton FL 33431

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Library of Congress Cataloging-in-Publication Data
Names: Altaç, Zekeriya, author.
Title: Numerical methods for scientists and engineers : with pseudocodes / Zekeriya Altaç.
Description: First edition. | Boca Raton : C&H/CRC Press, 2025. | Series: Chapman and Hall/CRC numerical analysis and scientific computing series | Includes bibliographical references and index.
Identifiers: LCCN 2024020497 (print) | LCCN 2024020498 (ebook) | ISBN 9781032754741 (hardback) | ISBN 9781032756424 (paperback) | ISBN 9781003474944 (ebook)
Subjects: LCSH: Numerical analysis--Textbooks. | Numerical analysis--Data processing--Textbooks.
Classification: LCC QA297 .A538 2025 (print) | LCC QA297 (ebook) | DDC 518--dc23/eng/20240522
LC record available at https://lccn.loc.gov/2024020497
LC ebook record available at https://lccn.loc.gov/2024020498

ISBN: 978-1-032-75474-1 (hbk)
ISBN: 978-1-032-75642-4 (pbk)
ISBN: 978-1-003-47494-4 (ebk)

DOI: 10.1201/9781003474944

Typeset in Latin Modern font
by KnowledgeWorks Global Ltd.

Publisher's note: This book has been prepared from camera-ready copy provided by the authors.

Access the Instructor Resources: www.routledge.com/9781032754741

Contents

Preface

This book is intended primarily as a core textbook for courses taught under the general title of "Numerical Methods." It is also suitable as a main or supplementary textbook for upper-level undergraduate or graduate courses, as well as as a reference book for practicing engineers.

The objective of the text can be summarized as follows: (i) Provide a strong theoretical background on numerical methods encountered in science and engineering education and beyond; (ii) Understand and apply the methods to practical problems; (iii) Develop the ability to choose a suitable method for a given problem; (iv) Gain the skill of converting numerical algorithms into running computer codes, emphasized *via* pseudocodes. I explain in detail the strategy implemented in the text to achieve the aforementioned objectives under the following main headings:

A. LEARNING OBJECTIVES

Each chapter begins with the "Learning Objectives," which briefly describe educational goals and competencies to be achieved. The theoretical and applied objectives make up the broader aims of the course. The learning objectives inform the students what they should master by the end of each chapter.

B. CONTENT

The text content is designed for a one-semester course. It consists of 11 chapters covering the fundamental methods as well as some advanced topics with significant reference values, which are left to the discretion of instructors. In the arrangement of the chapters, the contents of the subsequent chapters have been taken into account so that a student new to a subject can digest the course material chapter by chapter. In cases where there is little interdependence between the chapters, a sufficiently brief introduction to the pertinent topic is provided before it is covered in greater detail in the following chapter(s).

My thoughts on how to teach and design the course and its contents have been shaped over the past three decades while teaching engineering students. The S&E students want to see the relevance of this math course and its material in their professional education. Accordingly, in each chapter, the importance and application areas of the topics in S&E are briefly stated. Every method is kept as simple as possible, and proofs have only been included where they help students understand the underlying theory of the method better.

The S&E students learn best when they are motivated by solved examples and end-of-chapter exercises. Besides, this is a field in which problem solving by hand is essential to understanding the material, so a problem-solving perspective is adopted in this text. Immediately after a method is presented, its application is reinforced with example problems that comprehensively and thoroughly illustrate and explain the complexities and potential

difficulties of the method. Most examples and exercise problems consist of problems with known exact solutions, allowing a clear and concise assessment of the performance of the method. There are a total of 108 solved example problems, in which the solution is explained step-by-step and through error assessments. Furthermore, maximum effort has been put into preparing examples and end-of-chapter exercise problems from various disciplines, not only to demonstrate the relevance of the methods in S&E but also to illustrate the strengths and weaknesses of the presented methods. Each chapter ends with a Closure that summarizes the key features of the numerical methods presented.

C. ATTENTION BOXES

In order to grab the attention of students, key points, good practices, pitfalls, or warnings in connection with the application of a numerical method are highlighted throughout the text using the "attention boxes" with a hand pointing to information provided.

D. "PROS AND CONS" BOXES

Numerical methods devised for a specific mathematical problem differ from each other in objective, scope, degree of accuracy, amount of computation, and so on. In this regard, no single numerical method is suitable for handling every problem in its class. Consequently, there are usually several methods to solve a given problem numerically. For this reason, the basics of a numerical method and associated errors are presented in the theoretical part, and a clear and concise idea of when the method performs well or poorly is provided as well. This kind of knowledge is critical and necessary, even if one uses *modules* or *procedures* from available software packages. As a result, the ability to select a suitable method for a numerical task is crucial for the accurate and successful implementation of any numerical method. In this context, after a numerical method is presented, its advantages and disadvantages are presented in a "Pros and Cons Box." With this approach, by the end of the course, the students are expected to have gained the ability to question and understand the purpose and limitations of the methods or modules before using them.

E. USE OF PSEUDOCODES

All numerical algorithms in this text are presented as self-contained pseudomodules that can be converted and used in any program. The modules are written as simply as possible (with very few commands) so that translation to other programming platforms is as easy as possible. This approach blends the algorithms and methods with sufficient mathematical theory and supports easy implementation of the algorithms.

Why pseudocode? Today's engineers and scientists are expected to become proficient in one or more programming languages (C/C++, Python, Visual Basic, Fortran 90/95, etc.), as well as a number of software packages (MatLab, Mathematica, Maple, etc.) known as the *Computer* or *Symbolic Algebra System* (CAS or SAS). To be more general and inclusive, the text is not tied to any specific programming language or to any specific CAS software. This approach aims to teach numerical methods that students will encounter in their future courses or careers while enabling them to gain the ability to prepare and/or use well-structured programs in a variety of languages.

There are basically three main reasons why not to adopt popular "programming languages" or CAS software:

(1) Experience has shown over the last several decades that programming languages and/or software are highly transient in nature. They can become partially or completely

popular in one field (or course) or unpopular in another, or they can be replaced very quickly by other language or software in certain disciplines.

(2) Teaching numerical methods together with a particular programming language, or CAS, runs the risk of blinding the student to the underlying principles of the methods. Such practices inadvertently encourage students to simply copy, compile, and run the given source codes (or simply use or type the specified built-in *function*, *procedure*, or *module*) rather than attempt to understand the methods in depth, including their limitations or advantages. Furthermore, this attitude generally leads students to believe that any method can be applied to similar problems by utilizing readily available codes or software without having any knowledge or sufficient understanding of the relevant details.

(3) A majority of problems that scientists and engineers face today cannot be tackled by employing a single method (or ready-to-use modules). At some point, programming proficiency is required to implement a series of numerical methods within a broader computer program to achieve a more global computational objective. In this respect, the best way to learn numerical methods is to reinforce the application skills with computer programming while taking the "Numerical Methods" course.

The motivation for adopting a pseudocode approach is that the texts providing brief algorithms target an audience with some programming experience. A beginner-level undergraduate student without sufficient hands-on programming skills, in general, struggles to convert such algorithms into running computer programs. The pseudocodes adopted in the text mimic the full features of a real program, including input/output statements, structured coding, and line-by-line explanatory comments and annotations, so that translation to other programming languages is as simple as possible. By incorporating the pseudocode approach adopted here as a teaching tool, a student with minimum programming skills should be able to not only grasp the logic of the numerical method but also gain insight on how to implement it in a programming language of his or her choice.

The text contains a total of 96 pseudocodes. Every numerical algorithm is presented as a completely independent pseudomodule that can be converted and used with other programs. Some pseudocodes use one or more of the pseudomodules presented in prior sections or chapters. This text also puts forth a convention for writing pseudocodes since there is no common standard. However, basic structured programming statements and commands of common programming languages are adopted in this pseudocode convention, which is presented in the Appendix section under the heading "How to Read and Write Pseudocodes." With this as a reference, a student with an elementary programming background can easily follow, write, and convert the pseudocodes to any programming language s/he desires.

F. END-OF-CHAPTER "EXERCISE" PROBLEMS:

The text includes more than 1500 end-of-chapter exercise problems presented at the end of each chapter, organized according to sections. These problems emphasize the mechanics of the method and are designed to give practice in applying the algorithms presented. The majority of the exercises are made up of problems with true (exact) solutions, allowing for an accurate quantitative performance assessment of the methods. Some problems that are purely mathematical in nature are designed not only to reinforce methods through practice but also to demonstrate the strengths and weaknesses of a method. Additional applied or realistic problems are chosen from S&E disciplines to help reinforce understanding of methods and demonstrate their relevance in applied problems.

The exercises are designed to be assigned to students as self-study or homework prob-

lems. Some exercises do require computer programming to carry out calculations. Even in such cases, it is important for the students to solve most of the exercise problems or, at least, the first few steps by hand or by using spreadsheets to ensure full comprehension of the details of an algorithm before attempting to program it. Such exercise problems can also be used to validate the results of a computer program or to compare them with the results of other available software.

G. END-OF-CHAPTER "COMPUTER ASSIGNMENTS"

It is important for students to be able to not only select and use suitable modules (methods) from the pertinent software but also program the numerical algorithms in a language of their (or the instructor's) choice. To develop programming and parametric analysis skills, additional exercise problems 115 are provided as "Computer Assignments" at the end of each chapter, which require using and/or writing and running computer code (of the instructor's choice) and analyzing the results. The instructors may use or modify some exercises or computer assignment problems to allow methodological analysis and comparisons. By the end of the course, the student should realize and appreciate the importance of this skill in his profession.

A NOTE TO THE INSTRUCTOR

This author's experience and thoughts on teaching numerical methods were shaped as follows: applying the course contents effectively to practical problems requires theoretical knowledge as well as computational experience. The knowledge of how well or poorly a method will perform is extremely important and necessary. The student should be able to understand the purpose and limitations of a method or module to know whether it applies to his problem or not. Most problems that today's professionals deal with in practice require the use of several numerical methods and cannot be simply solved by adopting a standard module or subprogram. For such problems, students should be able to find solutions by putting together a number of suitable numerical methods and developing skills in adopting numerical methods for a new problem.

The course instructor has a significant influence on the direction of the course and what students can gain from it. In this context, this text empowers the instructors to utilize Excel and/or MatLab, Mathematica, MathCad, and similar CAS software. Besides implementing readily available modules or procedures specific to a particular algorithm, students can be directed to develop simple, well-structured programs to extend the base capabilities of such software environments. Alternatively, this strategy can be employed in programming languages such as Python, C/C++, Visual Basic, Fortran 90, and so on. Combined with realistic problems that require analysis, this approach allows students to be able to identify, select, and employ algorithms (or modules) for a specific task, allowing them to reinforce the numerical methods by experimenting with competing methods and helping them to explore their advantages and limitations.

I will be maintaining the webpage, set up specifically for this text to aid the students as well as instructors, which can be accessed at

https://github.com/zaltac/NumMethodsWPseudoCodes

It will be dynamic, and I will be posting modules on various programming platforms, PowerPoint presentations, solutions to additional exercises, errata, and more. I encourage

the instructors who adopt this text as a coursebook to provide me with feedback on how I can improve the materials that I will be sharing on this website.

A NOTE TO THE STUDENT

Numerical methods is a very different area of mathematics and certainly different from your previous math courses. In math courses, the tasks are clearly defined, with a very concrete notion of "the right answer." Here, we are concerned with computing approximations, and this involves a slightly different kind of thinking. We have to understand what we are approximating well enough to construct a reasonable approximation, and we have to be able to think clearly and logically enough to analyze the accuracy and performance of that approximation.

The pseudocodes are not intended to be programs with all possible routes or advanced programming techniques. They are designed in a simple way that will help you understand and implement the algorithms. Writing your own programs will help you understand how the method works. In this regard, the computational effort that you put in is an important part of learning numerical methods.

ACKNOWLEDGMENTS

I am indebted to a number of people for their encouragement and assistance during the preparation of this and previous texts. I would like to express my gratitude to all of my students and colleagues for their support and encouragement. Particular thanks are due to Necati MAHİR, Hasan Hüseyin ERKAYA, Nevzat KIRAÇ, Mesut TEKKALMAZ, Nihal UĞURLUBİLEK, and Zerrin SERT for proof-reading the Turkish or English versions of the manuscript. I would like to extend a special thanks to Gökçe Mehmet AY for reintroducing me to the LaTeX world. I would also like to thank Callum Fraser of CNC Press for his continued interest and advice in making this project possible.

I thank my mother Muzaffer, my siblings Mithat, Afitap, and Asuman, my daughter Delilah, and my son-in-law Mahmut for their unconditional love and support, as well as my grandchildren Efe and Meryem, who fill my heart with joy each and every day. Finally, I thank my wife Leslie for her love, enduring patience, and unconditional support, without which the book would probably not have been completed.

Zekeriya ALTAÇ
April, 2024.
Eskişehir, Turkiye

e-mail: altacz@gmail.com
https://github.com/zaltac/NumMethodsWPseudoCodes

Dedication

I dedicate this book to my beloved uncle Aydın BİNGÖL, whom I have always looked up to in all my endeavors. Without his support, I would not have been able to get my master's and doctoral degrees in the USA and to be academically where I am today. I will always be grateful to him.

Author Bios

The author, a professor of mechanical engineering at Eskişehir Osmangazi University, received his B.Sc. degree in 1983 from Istanbul Technical University in Turkey. He received a scholarship from the Ministry of Education of Turkey to study nuclear engineering in the USA. He attended Iowa State University and earned his M.Sc. (1986) and Ph.D. (1989) in Nuclear Engineering. Upon returning to Turkey, he worked as an assistant professor at the Mechanical Engineering Department of Anadolu University between 1989 and 1992 in Eskişehir, Turkey. Later, he joined the Department of Mechanical Engineering at Eskişehir Osmangazi University as an associate professor in 1993 and received a tenure-track position in 1998, where he is currently working. His areas of interest and expertise are computational nuclear reactor physics, computational fluid mechanics and heat transfer, and numerical (computational) methods in general. He is the author of over 100 journal articles, conference and research papers, books, and book translations. The author has books in Turkish on computer programming (BASIC, FORTRAN) and the application of CFD software (ANSYS, FLUENT). Most of his research deals with developing efficient numerical methods and/or employing computational techniques in practical scientific and engineering problems. He served as an advisor at the Von Karman Institute Technical Advisory Committee (VKI-TAC) in Brussels, Belgium, between 2009 and 2011. He has also served as Eskişehir Osmangazi University's vice president (2007-2011), as head of the Mechanical Engineering Department (2012-2021), and held positions on numerous conference, symposium, and academic committees, such as the engineering college board and university senate.

CHAPTER 1

Numerical Algorithms and Errors

LEARNING OBJECTIVES

After completing this chapter, you should be able to

- state the purpose and benefits of numerical methods in science and engineering;
- read and apply pseudocode conventions to understand and implement algorithms;
- understand how numerical data is managed in computers;
- define absolute/relative errors and use them in error reporting;
- understand and apply the error propagation formula in error analysis;
- realize the computer representation of numbers in different bases;
- distinguish the difference between chopping and rounding;
- determine and apply the concept of significant digits;
- state the difference between *accuracy* and *precision*;
- discuss the effects of rounding on floating-point arithmetic operations;
- understand the concept of *loss of significance* and how to avoid it;
- determine the convergence rate of sequences;
- explain the significance of the *condition* of a problem and the *stability* of a method;
- determine the truncation error of a series approximation;
- apply the mean value theorems to practical problems;
- investigate and identify the truncation error of a positive and alternating series.

NUMERICAL algorithms were first developed for hand calculations in the 19^{th} and early 20^{th} centuries to tackle mathematical problems such as solving small systems of linear equations, inverting matrices, approximating integrals, and solving ordinary differential equations. Early numerical solutions were obtained manually until the introduction of the computer after World War II. Rapid technological advances led to the development of the high-speed computers we use today. As a result, making use of the computer's capability to perform long sequences of calculations has brought about advances and further technological developments in many fields, and computers and mathematical sciences naturally became inseparable.