JOURNAL OF NUMERICAL ANALYSIS AND APPROXIMATION THEORY

J. Numer. Anal. Approx. Theory, vol. 54 (2025) no. 1, pp. 191–193,

`ictp.acad.ro/jnaat`

BOOK REVIEWS

David L. Chopp, *Introduction to High Performance Scientific Computing*, SIAM, Philadelphia, 2019, ISSN: 9781611975642, ISSN (print): 9781611975635.

Based on a 10-week course taught at Northwestern University, this book offers a practical and accessible introduction to scientific computing. It is designed for students with a basic programming background and aims to teach how to effectively use modern computing tools for scientific problems. Each part of the book focuses on a different programming model, making it flexible for different course structures and hardware settings.

The book begins with a course in the C programming language, chosen for its simplicity, portability, and efficiency. While those familiar with *C++* will also find it accessible, thanks to helpful notes and equivalent examples, the reader is introduced from the basics, starting with program structure, variables, library usage, and other introductory elements.

Parallel computing is introduced in the next two parts of the book. Shared-memory parallelism is covered first, using OpenMP—a lightweight approach well-suited for desktop or multi-core machines. This section begins with introductory conceptual notions on parallelism, such as threads, loop subdivision, and various useful clauses. At the end of this part, there are suggestions for projects to be implemented using parallelism, including the Allen–Cahn equation, Euler equations, Stokes flow, and others. In contrast to this approach—which is suitable for a single computer with perhaps multiple processors—the next section covers distributed-memory parallelism using MPI. This model is intended for large-scale computations across multiple nodes, such as those used in high-performance clusters like Northwestern's Quest system. This section also includes suggested projects to be implemented using MPI.

GPU computing is treated in the next two parts of the book. CUDA is used to teach GPU programming on NVIDIA hardware, showing how data-parallel tasks can be accelerated significantly. OpenCL is presented as a more portable, vendor-neutral alternative, although it is somewhat less refined than CUDA. The final part of the book connects all these techniques by applying them to real scientific problems: differential equations, finite difference methods as well as pseudospectral methods.

The author's approach emphasizes hands-on learning, encouraging readers to experiment with editors, compilers, and profilers. Programming is not taught for its own sake, but as a tool to solve real numerical problems. Exercises throughout the book help solidify both technical skills and mathematical understanding, particularly for topics like numerical methods and mathematical modeling.

The modular structure makes the book suitable for various teaching paths. For example, a 10-week course might include C, MPI, and CUDA; a shared-memory only course could swap MPI for OpenMP; a longer course might also include OpenCL and more extensive work with applications. Supporting material, including code in both C and C++, is available online, and external references like cppreference.com are suggested for further reading.

Overall, the book provides a concise yet comprehensive pathway into scientific computing, combining programming techniques with practical applications. Its structure, clarity, and focus on real-world use cases make it especially useful for students, instructors, and professionals looking to build a strong foundation in this area.

Andrei Stan

Samuel D. Conte, Carl De Boor, *Elementary numerical analysis: an algorithmic approach*, SIAM, Philadelphia, 2018, XXIV + 454 pp., ISBN 978-1-61197-519-2 (paperback), eISBN 978-1-61197-520-8 (ebook). Part of the Classics in Applied Mathematics series.

Published in the SIAM Series on Classics in Applied Mathematics, this book brings together important topics on elementary numerical analysis. The book presents a detailed and comprehensive introduction to scientific computing, combining theoretical knowledge with practical implementation. It adopts a rigorous yet accessible approach, seamlessly connecting foundational theories with illustrative examples, algorithms, and programming applications. The overall picture of the book includes the rich experience of the authors in mathematics and computer science. As a former manager of the Math and Programming Department of the Aerospace Corporation, head of the Mathematics and Computer and Programming Departments at TRW Inc and Professor of Computer Science and Mathematics at Purdue University, Samuel D. Conte brings together his academic and scientific background with Carl de Boor's experience in scientific computing. Professor de Boor is the author of *A Practical Guide to Splines* and co-author of *Box Splines*. He is the recipient of the 2003 National Medal of Science in Mathematics and Computer Science for his contributions in scientific computing.

The book contains 9 chapters, an Appendix of 2 topics, a Preface followed by an Errata, Bibliography and Index. The entire content is organized in an accessible manner and the Preface contains a short presentations of the chapters. Throughout the book, a great number of exercises and problems are proposed. These exercises and problems have different degrees of difficulty, being marked accordingly at the end of each chapter. Also, a lot of examples and solved exercises are included in each chapter and for some of them there are MATLAB programs at the end of the book.

In Chapter 1 the authors discuss some mathematical preliminaries and different methods for representing numbers on computers (integers, fractions). The errors introduced by these representations are studied together with the sources of various type of computational errors and their propagation in algorithms.

Chapter 2 is dedicated to polynomial interpolation. Since the construction, representation and evaluation of polynomials is very easy, they are used in the solution of equations and in the approximation of functions (integrals, derivatives). An important part of this chapter is devoted to the study of divided-differences table.

Another important topic that is considered in Chapter 3 is the study of the solutions of nonlinear equations. In this chapter, the authors consider various iterative methods for finding approximations to simple roots of the equation $f(x) = 0$ in different forms. This chapter contains a survey of iterative methods, a section dedicated to fixed-point iteration method and notion related to convergence of these methods.

Chapter 4 addresses the field of matrices and systems of linear equations. After a substantial part dedicated to results and properties of matrices, this chapter continues with important notions related to numerical solutions of linear systems. This part includes important remarks on Gauss elimination, LU decomposition (Choleski algorithm), conditioning number for a matrix and study of the error.

In Chapter 5 the authors include a brief discussion of systems of equations and unconstrained optimization. The first section contains basic results in optimization and steepest descent method (theory, algorithms, examples and exercises). The next two sections are dedicated to Newton's method, fixed-point interation and relaxation method, respectively.

Chapter 6 discusses the fundamental idea of function approximation, which serves two main purposes: simplifying complicated functions for easier mathematical operations and reconstructing functions from limited data. The text introduces key classes of approximating functions, including algebraic polynomials, trigonometric polynomials, and piecewise-polynomial functions, focusing on finding the best possible approximation within each category.

Together with all topics presented above, the authors also bring to the reader's attention notions of numerical differentiation and integration. This chapter contains the most important rules that are used in this context in order to obtain good approximations for derivatives, respectively integrals of real functions.

Chapter 8 is dedicated to studying methods for solving differential equations. As the authors says, many problems in engineering and science can be formulate in terms of differential equations. Unfortunately, most of the equations that arise in practice cannot be solved analytically, so we resort to numerical methods. The chapter presents several methods for solving differential equations, with special emphasis on Runge-Kutta and Adams-Moulton methods.

The last chapter generalizes the result presented above and addresses problems in which the initial conditions are specified at more than one point. In this part the authors study boundary-value problems in ordinary differential equations. The most important methods used here are: the method of finite differences, the so-called shooting method and the method of collocation.

Certainly, the book is extremely useful for undergraduate students in engineering, mathematics and science, in particular, computer science. It is an important material that includes - in addition to standard notions of elementary numerical analysis - several examples, solved exercises and MATLAB programs.

Eduard Grigoriciuc