Numerical Analysis for Engineers and Scientists

Striking a balance between theory and practice, this graduate-level text is perfect for students in the applied sciences. The author provides a clear introduction to the classical methods, how they work and why they sometimes fail. Crucially, he also demonstrates how these simple and classical techniques can be combined to address difficult problems. Many worked examples and sample programs are provided to help the reader make practical use of the subject material. Further mathematical background, if required, is summarized in an appendix.

Topics covered include classical methods for linear systems, eigenvalues, interpolation and integration, ODEs and data fitting, and also more modern ideas such as adaptivity and stochastic differential equations.

G. Miller is a Professor in the Department of Chemical Engineering and Materials Science at University of California, Davis.

Cambridge University Press
978-1-107-02108-2 - Numerical Analysis for Engineers and Scientists
G. Miller
Frontmatter
More information

Numerical Analysis for Engineers and Scientists

G. MILLER
Department of Chemical Engineering and Materials Science
University of California, Davis

CAMBRIDGE UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Published in the United States of America by Cambridge University Press, New York

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107021082

First published 2014

Printed in the United Kingdom by CPI Group Ltd, Croydon CR0 4YY

A catalogue record for this publication is available from the British Library

ISBN 978-1-107-02108-2 Hardback

Contents

Preface

This book is an introduction to numerical analysis: the solution of mathematical problems using numerical algorithms. Typically these algorithms are implemented with computers. To solve numerically a problem in science or engineering one is typically faced with four concerns:

1. How can the science/engineering problem be posed as a mathematical problem?
2. How can the mathematical problem be solved at all, using a computer?
3. How can it be solved accurately?
4. How can it be solved quickly?

The first concern comes from the science and engineering disciplines, and is outside the scope of this book. However, there are many practical examples and problems drawn from engineering, chemistry, and economics applications.

The focus of most introductory texts on numerical methods is, appropriately, concern #2, and that is also the main emphasis here. Accordingly, a number of different subjects are described that facilitate solution of a wide array of science and engineering problems.

Accuracy, concern #3, deals with numerical error and approximation error. There is a brief introductory chapter on error that presents the main ideas. Throughout the remainder of this book, algorithm choices and implementation details that affect accuracy are described. For the most part, where a claim of accuracy is made an example is given to illuminate the point, and to show how such claims can be tested.

The speed of computational methods, concern #4, is addressed by emphasizing two aspects of algorithm design that directly impact performance in a desktop environment – rates of convergence and operation count, and by introducing adaptive algorithms which use resources judiciously. Numerous examples are provided to illustrate rates of convergence. Modern high-performance algorithms are concerned also with cache, memory, and communication latency, which are not addressed here.

In some circles there is a tendency, bolstered by Moore's law [166], to suppose that accuracy and speed are not terribly important in algorithm design. In two years, one can likely buy a computer that is twice as fast as the best available today. So, if speed is important it might be best to acquire a faster platform. Likewise, a faster, more capable, computer could employ arbitrarily high precision to overcome any of today's accuracy problems. However, in a given computing environment the fast algorithm will always out-perform the slow ones, so Moore's law does not affect the relative performance.

Similarly, one can always implement a more accurate algorithm with higher precision, and for given precision the more accurate algorithm will always prevail.

The material covered in this book includes representative algorithms that are commonly used. Most are easily implemented or tested with pencil, paper, and a simple hand calculator. It is interesting to note that most of the methods that will be described predate modern computers, so their implementation by hand is not at all unreasonable. In fact, until the 1950s the term “computer” referred to a person whose profession was performing calculations by hand or with slide rules. To emphasize the antiquity of some of these ideas, and to give proper recognition to the pioneers that discovered them, I attempt to provide references to the original works.

Texts on numerical analysis and numerical methods range from very practical to very theoretical, and in this one I hope to strike a balance. On the practical side, there are numerous worked solutions and code examples. The code examples are intended to be a compromise between pseudocode and production code – functional and readable, but not state of the art. I hope the interested reader will see the similarity between the equations in the text and the `C++` code to get a better appreciation of the logic, and the accessibility of these methods (i.e., if I can do it, so can you). These codes are available online for download at `http://www.cambridge.org/9781107021082`. On the theoretical side, the mathematical approaches used to derive and explain numerical algorithms are different from those a typical engineering student will have encountered in calculus and analytical partial differential equations. This is both interesting and useful, and I have included some in an informal way. There are no theorems, but the logic is displayed through equations and text. Some of the mathematical background needed to understand these concepts is summarized in an appendix.

This book grew from class notes developed over a dozen years of teaching numerical methods to engineers at both undergraduate and graduate levels. In a 10-week undergraduate course one or two examples from each of the first 10 chapters can be discussed to give an overview of the field and to equip the students with some numerical problem solving skills. In a 20-week graduate course, most of the material can be covered with reasonable depth.

I thank the students of EAD 115 and EAD 210 who all contributed to the development of this book through their engagement and feedback over the years. In particular, I thank Bakytzhan Kallemov for helping to develop the chapter on stochastic methods, and Mehdi Vahab for improving the sample codes. I am especially grateful to my wife Carolyn for her support and encouragement.