

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](#)

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
978-1-107-02108-2 - Numerical Analysis for Engineers and Scientists
G. Miller
Frontmatter
[More information](#)

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

© G. Miller 2014

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

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

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

Contents

	<i>Preface</i>	<i>page ix</i>
1	Numerical error	1
	1.1 Types of error	1
	1.2 Floating point numbers	1
	1.3 Algorithms and error	6
	1.4 Approximation error vs. algorithm error	13
	1.5 An important example	16
	1.6 Backward error	17
	Problems	19
2	Direct solution of linear systems	22
	2.1 Gaussian elimination	23
	2.2 Pivot selection	30
	2.3 Errors in Gaussian elimination	36
	2.4 Householder reduction	43
	2.5 Cholesky decomposition	49
	2.6 The residual correction method	52
	Problems	54
3	Eigenvalues and eigenvectors	56
	3.1 Gerschgorin's estimate	57
	3.2 The power method	62
	3.3 The QR algorithm	70
	3.4 Singular value decomposition	81
	3.5 Hyman's method	89
	Problems	91
4	Iterative approaches for linear systems	93
	4.1 Conjugate gradient	93
	4.2 Relaxation methods	103
	4.3 Jacobi	105
	4.4 Irreducibility	109
	4.5 Gauss–Seidel	111

	4.6 Multigrid	115
	Problems	123
5	Interpolation	125
	5.1 Modified Lagrange interpolation and the barycentric form	128
	5.2 Neville's algorithm	129
	5.3 Newton	131
	5.4 Hermite	136
	5.5 Discrete Fourier transform	138
	Problems	148
6	Iterative methods and the roots of polynomials	153
	6.1 Convergence and rates	153
	6.2 Bisection	155
	6.3 Regula falsi	157
	6.4 The secant method	159
	6.5 Newton–Raphson	162
	6.6 Roots of a polynomial	168
	6.7 Newton–Raphson on the complex plane	172
	6.8 Bairstow's method	175
	6.9 Improving convergence	179
	Problems	180
7	Optimization	182
	7.1 1D: Bracketing	182
	7.2 1D: Refinement by interpolation	183
	7.3 1D: Refinement by golden section search	184
	7.4 nD: Variable metric methods	185
	7.5 Linear programming	191
	7.6 Quadratic programming	201
	Problems	211
8	Data fitting	213
	8.1 Least squares	213
	8.2 An application to the Taylor series	217
	8.3 Data with experimental error	219
	8.4 Error in x and y	225
	8.5 Nonlinear least squares	230
	8.6 Fits in other norms	230
	8.7 Splines	235
	Problems	241

9	Integration	243
	9.1 Newton–Cotes	243
	9.2 Extrapolation	252
	9.3 Adaptivity	257
	9.4 Gaussian quadrature	259
	9.5 Special cases	271
	Problems	273
10	Ordinary differential equations	275
	10.1 Initial value problems I: one-step methods	275
	10.2 Initial value problems II: multistep methods	278
	10.3 Adaptivity	287
	10.4 Boundary value problems	292
	10.5 Stiff systems	298
	Problems	300
11	Introduction to stochastic ODEs	302
	11.1 White noise and the Wiener process	303
	11.2 Itô and Stratonovich calculus	306
	11.3 Itô’s formula	308
	11.4 The Itô–Taylor series	309
	11.5 Orders of accuracy	311
	11.6 Strong convergence	314
	11.7 Weak convergence	318
	11.8 Modeling	320
	Problems	324
12	A big integrative example	326
	12.1 The Schrödinger equation	326
	12.2 Gaussian basis functions	337
	12.3 Results I: H ₂	341
	12.4 Angular momentum	342
	12.5 Rys polynomials	345
	12.6 Results II: H ₂ O	349
Appendix A	Mathematical background	353
	A.1 Continuity	353
	A.2 Triangle inequality	354
	A.3 Rolle’s theorem	354
	A.4 Mean value theorem	355
	A.5 Geometric series	356
	A.6 Taylor series	357
	A.7 Linear algebra	361
	A.8 Complex numbers	369

Appendix B	Sample codes	371
B.1	Utility routines	371
B.2	Gaussian elimination	375
B.3	Householder reduction	380
B.4	Cholesky reduction	386
B.5	The QR method with shifts for symmetric real matrices	388
B.6	Singular value decomposition	393
B.7	Conjugate gradient	400
B.8	Jacobi, Gauss–Seidel, and multigrid	402
B.9	Cooley–Tukey FFT	405
B.10	Variable metric methods	408
B.11	The simplex method for linear programming	413
B.12	Quadratic programming for convex systems	420
B.13	Adaptive Simpson’s rule integration	426
B.14	Adaptive Runge–Kutta ODE example	428
B.15	Adaptive multistep ODE example	430
B.16	Stochastic integration and testing	434
B.17	Big example: Hartree–Fock–Roothaan	438
	<i>Solutions</i>	454
	<i>References</i>	555
	<i>Index</i>	567

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.