Texts in Computational Science and Engineering 2

Editors

For further volumes:
http://www.springer.com/series/5151

Alfio Quarteroni • Fausto Saleri •
Paola Gervasio

Scientific Computing with MATLAB and Octave

Fourth Edition

Springer

Alfio Quarteroni
MATHICSE-CMCS
Ecole Polytechnique Fédérale
de Lausanne
Lausanne
Switzerland

Fausto Saleri (1965-2007)
MOX - Politecnico di Milano
Milano
Italy

Paola Gervasio
DICATAM
Università degli Studi di Brescia
Brescia
Italy

ISSN 1611-0994
ISBN 978-3-642-45366-3 ISBN 978-3-642-45367-0 (eBook)
DOI 10.1007/978-3-642-45367-0
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014932425

Mathematics Subject Classification (2010): 65-01, 68U01, 68N15

The picture on the cover shows a re-entrant electrical wave on a slab of homogeneous excitable medium, exhibiting spiral turbulence and spatiotemporal chaos. Computation by Ricardo Ruiz-Baier, IST, University of Lausanne, CH.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To the memory of
Fausto Saleri

Preface

Preface to the First Edition

This textbook is an introduction to Scientific Computing. We will illustrate several numerical methods for the computer solution of certain classes of mathematical problems that cannot be faced by paper and pencil. We will show how to compute the zeros or the integrals of continuous functions, solve linear systems, approximate functions by polynomials and construct accurate approximations for the solution of differential equations.

With this aim, in Chapter 1 we will illustrate the rules of the game that computers adopt when storing and operating with real and complex numbers, vectors and matrices.

In order to make our presentation concrete and appealing we will adopt the programming environment MATLAB® [1] as a faithful companion. We will gradually discover its principal commands, statements and constructs. We will show how to execute all the algorithms that we introduce throughout the book. This will enable us to furnish an immediate quantitative assessment of their theoretical properties such as stability, accuracy and complexity. We will solve several problems that will be raised through exercises and examples, often stemming from specific applications.

Several graphical devices will be adopted in order to render the reading more pleasant. We will report in the margin the MATLAB command along side the line where that command is being introduced for the first time. The symbol  will be used to indicate the presence of exercises, the symbol  to indicate the presence of a MATLAB program, while

[1] MATLAB is a trademark of TheMathWorks Inc., 24 Prime Park Way, Natick, MA 01760, Tel: 001+508-647-7000, Fax: 001+508-647-7001.

the symbol  will be used when we want to attract the attention of the reader on a critical or surprising behavior of an algorithm or a procedure. The mathematical formulae of special relevance are put within a frame. Finally, the symbol  indicates the presence of a display panel summarizing concepts and conclusions which have just been reported and drawn.

At the end of each chapter a specific section is devoted to mentioning those subjects which have not been addressed and indicate the bibliographical references for a more comprehensive treatment of the material that we have carried out.

Quite often we will refer to the textbook [QSS07] where many issues faced in this book are treated at a deeper level, and where theoretical results are proven. For a more thorough description of MATLAB we refer to [HH05]. All the programs introduced in this text can be downloaded from the web address

`mox.polimi.it/qs`

No special prerequisite is demanded of the reader, with the exception of an elementary course of Calculus.

However, in the course of the first chapter, we recall the principal results of Calculus and Geometry that will be used extensively throughout this text. The less elementary subjects, those which are not so necessary for an introductory educational path, are highlighted by the special symbol .

We express our thanks to Thanh-Ha Le Thi from Springer-Verlag Heidelberg, and to Francesca Bonadei and Marina Forlizzi from Springer-Italia for their friendly collaboration throughout this project. We gratefully thank Prof. Eastham of Cardiff University for editing the language of the whole manuscript and stimulating us to clarify many points of our text.

Milano and Lausanne — *Alfio Quarteroni*
May 2003 — *Fausto Saleri*

Preface to the Second Edition

In this second edition we have enriched all the Chapters by introducing several new problems. Moreover, we have added new methods for the numerical solution of linear and nonlinear systems, the eigenvalue computation and the solution of initial-value problems. Another relevant improvement is that we also use the Octave programming environment. Octave is a reimplementation of part of MATLAB which includes many numerical facilities of MATLAB and is freely distributed under the GNU General Public License.

Throughout the book, we shall often make use of the expression "MATLAB command": in this case, MATLAB should be understood as the *language* which is the common subset of both programs MATLAB and Octave. We have striven to ensure a seamless usage of our codes and programs under both MATLAB and Octave. In the few cases where this does not apply, we shall write a short explanation notice at the end of each corresponding section.

For this second edition we would like to thank Paola Causin for having proposed several problems, Christophe Prud´homme, John W. Eaton and David Bateman for their help with Octave, and Silvia Quarteroni for the translation of the new sections. Finally, we kindly acknowledge the support of the Poseidon project of the Ecole Polytechnique Fédérale de Lausanne.

Lausanne and Milano
May 2006

Alfio Quarteroni
Fausto Saleri

Preface to the Third Edition

This third edition features a complete revisitation of the whole book, many improvements in style and content to all the chapters, as well as a substantial new development of those chapters devoted to the numerical approximation of boundary-value problems and initial-boundary-value problems. We remind the reader that all the programs introduced in this text can be downloaded from the web address

`mox.polimi.it/qs`

Lausanne, Milano and Brescia
March 2010

Alfio Quarteroni
Paola Gervasio

Preface to the Fourth Edition

The fourth edition features the addition of a new chapter on numerical optimization of both univariate and multivariate functions in which several methods are presented, discussed and analyzed.

For unconstrained minimization, we consider derivative free methods, descent (or line search) methods, and trust region methods.

For constrained minimization we restrict our discussion to penalization methods and augmented Lagrangian methods.

As for the other chapters of this book, also this new chapter is supported by examples, exercises and programs written in both MATLAB and Octave environments.

The addition of this chapter made it necessary a renumbering of several other chapters with respect to the previous editions. Moreover, new sections have been added in some other chapters.

Finally we remind the reader that all programs presented in this book can be downloaded from the web address

`http://mox.polimi.it/qs`

Lausanne, Milano and Brescia — *Alfio Quarteroni*
December 2013 — *Paola Gervasio*

Contents

1 What can't be ignored 1
1.1 The MATLAB and Octave environments 1
1.2 Real numbers 3
1.2.1 How we represent them 3
1.2.2 How we operate with floating-point numbers 6
1.3 Complex numbers 8
1.4 Matrices 10
1.4.1 Vectors 14
1.5 Real functions 16
1.5.1 The zeros 19
1.5.2 Polynomials 20
1.5.3 Integration and differentiation 22
1.6 To err is not only human 25
1.6.1 Talking about costs 28
1.7 The MATLAB language 30
1.7.1 MATLAB statements 32
1.7.2 Programming in MATLAB 34
1.7.3 Examples of differences between MATLAB and Octave languages 38
1.8 What we haven't told you 38
1.9 Exercises 39

2 Nonlinear equations 41
2.1 Some representative problems 41
2.2 The bisection method 43
2.3 The Newton method 47
2.3.1 How to terminate Newton's iterations 50
2.4 The secant method 51
2.5 Systems of nonlinear equations 52

2.6 Fixed point iterations 56
2.6.1 How to terminate fixed point iterations 62
2.7 Acceleration using Aitken method 63
2.8 Algebraic polynomials 67
2.8.1 Hörner's algorithm 68
2.8.2 The Newton-Hörner method 70
2.9 What we haven't told you 72
2.10 Exercises 74

3 Approximation of functions and data 77
3.1 Some representative problems 77
3.2 Approximation by Taylor's polynomials 79
3.3 Interpolation 80
3.3.1 Lagrangian polynomial interpolation 81
3.3.2 Stability of polynomial interpolation 86
3.3.3 Interpolation at Chebyshev nodes 87
3.3.4 Barycentric interpolation formula 90
3.3.5 Trigonometric interpolation and FFT 93
3.4 Piecewise linear interpolation 98
3.5 Approximation by spline functions 100
3.6 The least-squares method 104
3.7 What we haven't told you 108
3.8 Exercises 110

4 Numerical differentiation and integration 113
4.1 Some representative problems 113
4.2 Approximation of function derivatives 115
4.3 Numerical integration 117
4.3.1 Midpoint formula 118
4.3.2 Trapezoidal formula 120
4.3.3 Simpson formula 121
4.4 Interpolatory quadratures 123
4.5 Simpson adaptive formula 127
4.6 Monte Carlo Methods for Numerical Integration 131
4.7 What we haven't told you 133
4.8 Exercises 134

5 Linear systems 137
5.1 Some representative problems 137
5.2 Linear system and complexity 142
5.3 The LU factorization method 143
5.4 The pivoting technique 154
5.4.1 The *fill-in* of a matrix 157
5.5 How accurate is the solution of a linear system? 158
5.6 How to solve a tridiagonal system 162

5.7 Overdetermined systems 163
5.8 What is hidden behind the MATLAB command \ 166
5.9 Iterative methods 168
5.9.1 How to construct an iterative method 169
5.10 Richardson and gradient methods 174
5.11 The conjugate gradient method 177
5.12 When should an iterative method be stopped? 180
5.13 To wrap-up: direct or iterative? 182
5.14 What we haven't told you 188
5.15 Exercises 188

6 Eigenvalues and eigenvectors 193
6.1 Some representative problems 194
6.2 The power method 196
6.2.1 Convergence analysis 199
6.3 Generalization of the power method 201
6.4 How to compute the shift 203
6.5 Computation of all the eigenvalues 206
6.6 What we haven't told you 209
6.7 Exercises 210

7 Numerical optimization 213
7.1 Some representative problems 214
7.2 Unconstrained optimization 217
7.3 Derivative free methods 219
7.3.1 Golden section and quadratic interpolation methods 219
7.3.2 Nelder and Mead method 223
7.4 The Newton method 227
7.5 Descent (or line search) methods 228
7.5.1 Descent directions 229
7.5.2 Strategies for choosing the steplength α_k 231
7.5.3 The descent method with Newton's directions 237
7.5.4 Descent methods with quasi-Newton directions 238
7.5.5 Gradient and conjugate gradient descent methods 240
7.6 Trust region methods 242
7.7 The nonlinear least squares method 248
7.7.1 Gauss-Newton method 249
7.7.2 Levenberg-Marquardt's method 252
7.8 Constrained optimization 253
7.8.1 The penalty method 259
7.8.2 The augmented Lagrangian method 264
7.9 What we haven't told you 267
7.10 Exercises 268

8 Ordinary differential equations 271
8.1 Some representative problems 271
8.2 The Cauchy problem 274
8.3 Euler methods 275
8.3.1 Convergence analysis 278
8.4 The Crank-Nicolson method 282
8.5 Zero-stability 284
8.6 Stability on unbounded intervals 286
8.6.1 The region of absolute stability 289
8.6.2 Absolute stability controls perturbations 290
8.6.3 Stepsize adaptivity for the forward Euler method 297
8.7 High order methods 300
8.8 The predictor-corrector methods 305
8.9 Systems of differential equations 307
8.10 Some examples 313
8.10.1 The spherical pendulum 313
8.10.2 The three-body problem 317
8.10.3 Some stiff problems 319
8.11 What we haven't told you 325
8.12 Exercises 326

9 Numerical approximation of boundary-value problems 329
9.1 Some representative problems 330
9.2 Approximation of boundary-value problems 332
9.2.1 Finite difference approximation of the one-dimensional Poisson problem 333
9.2.2 Finite difference approximation of a convection-dominated problem 336
9.2.3 Finite element approximation of the one-dimensional Poisson problem 337
9.2.4 Finite difference approximation of the two-dimensional Poisson problem 341
9.2.5 Consistency and convergence of finite difference discretization of the Poisson problem 347
9.2.6 Finite difference approximation of the one-dimensional heat equation 348
9.2.7 Finite element approximation of the one-dimensional heat equation 352
9.3 Hyperbolic equations: a scalar pure advection problem 355
9.3.1 Finite difference discretization of the scalar transport equation 357

9.3.2 Finite difference analysis for the scalar transport equation . . . 359
9.3.3 Finite element space discretization of the scalar advection equation . . . 366
9.4 The wave equation . . . 367
9.4.1 Finite difference approximation of the wave equation . . . 369
9.5 What we haven't told you . . . 373
9.6 Exercises . . . 374

10 Solutions of the exercises . . . 377
10.1 Chapter 1 . . . 377
10.2 Chapter 2 . . . 380
10.3 Chapter 3 . . . 385
10.4 Chapter 4 . . . 389
10.5 Chapter 5 . . . 394
10.6 Chapter 6 . . . 401
10.7 Chapter 7 . . . 404
10.8 Chapter 8 . . . 411
10.9 Chapter 9 . . . 422

References . . . 429

Index . . . 435

Index of MATLAB and Octave programs

All the programs introduced in this text can be downloaded from
http://mox.polimi.it/qs

2.1 **bisection**: bisection method . . . 45
2.2 **newton**: Newton method . . . 50
2.3 **newtonsys**: Newton method for nonlinear systems . . . 53
2.4 **aitken**: Aitken method . . . 65
2.5 **horner**: synthetic division algorithm . . . 69
2.6 **newtonhorner**: Newton-Hörner method . . . 71
3.1 **barycentric**: barycentric interpolation . . . 92
3.2 **cubicspline**: interpolating cubic spline . . . 101
4.1 **midpointc**: composite midpoint quadrature formula . . . 120
4.2 **simpsonc**: composite Simpson quadrature formula . . . 122
4.3 **simpadpt**: adaptive Simpson formula . . . 130
5.1 **lugauss**: Gauss LU factorization . . . 149
5.2 **itermeth**: general iterative method . . . 171
6.1 **eigpower**: power method . . . 198
6.2 **invshift**: inverse power method with shift . . . 202
6.3 **gershcircles**: Gershgorin circles . . . 203
6.4 **qrbasic**: method of QR iterations . . . 207
7.1 **golden**: golden section method . . . 221
7.2 **backtrack**: backtracking strategy . . . 234
7.3 **descent**: descent method . . . 235
7.4 **trustregion**: trust region method . . . 246
7.5 **gaussnewton**: Gauss-Newton method . . . 250
7.6 **penalty**: penalty method . . . 261
7.7 **auglagrange**: augmented Lagrangian method . . . 265
8.1 **feuler**: forward Euler method . . . 276
8.2 **beuler**: backward Euler method . . . 277

8.3 **cranknic**: Crank-Nicolson method 283
8.4 **predcor**: predictor-corrector method 306
8.5 **feonestep**: one step of the forward Euler method 307
8.6 **beonestep**: one step of the backward Euler method 307
8.7 **cnonestep**: one step of the Crank-Nicolson method 307
8.8 **newmark**: Newmark method 312
8.9 **fvinc**: forcing term for the spherical pendulum problem 316
8.10 **threebody**: forcing term for the simplified three body system 318
9.1 **bvp**: approximation of a two-point diffusion-convection-reaction problem by the finite difference method 335
9.2 **poissonfd**: approximation of the Poisson problem with Dirichlet boundary data by the five-point finite difference method 345
9.3 **heattheta**: θ-method for the one-dimensional heat equation 350
9.4 **newmarkwave**: Newmark method for the wave equation . . . 369
10.1 **gausslegendre**: Gauss-Legendre composite quadrature formula, with $n = 1$ 391
10.2 **rk2**: Heun (or RK2) method 414
10.3 **rk3**: explicit Runge-Kutta method of order 3 416
10.4 **neumann**: numerical solution of a Neumann boundary-value problem 423
10.5 **hyper**: Lax-Friedrichs, Lax-Wendroff and upwind schemes . . . 427