

Editors

Timothy J. Barth
Michael Griebel
David E. Keyes
Risto M. Nieminen
Dirk Roose
Tamar Schlick

Aslak Tveito • Hans Petter Langtangen
Bjørn Frederik Nielsen • Xing Cai

Elements of Scientific Computing

With 88 Figures and 18 Tables

 Springer

Aslak Tveito
Hans Petter Langtangen
Xing Cai
Center for Biomedical Computing
Simula Research Laboratory
P.O. Box 134
1325 Lysaker
Norway
aslak@simula.no
hpl@simula.no
xingcai@simula.no

and

Department of Informatics
University of Oslo
P.O. Box 1080 Blindern
0316 Oslo
Norway

Bjørn Frederik Nielsen
Department of Mathematical
Sciences and Technology
Norwegian University of Life Sciences
P.O. Box 5003
1432 Ås
Norway
bjorn.f.nielsen@umb.no

and

Center for Biomedical Computing
Simula Research Laboratory
P.O. Box 134
1325 Lysaker
Norway

Aslak Tveito has received financial support from the Norwegian Non-fiction Literature Fund.

ISSN 1611-0994

ISBN 978-3-642-11298-0 e-ISBN 978-3-642-11299-7

DOI 10.1007/978-3-642-11299-7

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010936979

Mathematics Subject Classification (2010): 97-01, 97M10, 97N80

© Springer-Verlag Berlin Heidelberg 2010

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: deblik, Berlin

Printed on acid-free paper

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

Preface

Science used to be experiments and theory; now it is experiments, theory, and computations. The computational approach to understanding nature and technology is currently expanding in many fields, such as physics, mechanics, geophysics, astrophysics, chemistry, biology, and most engineering disciplines. The computational methods used in these branches are very similar, and this book is a first introduction to such methods. Many books have been written on the subject. The present text aims to provide a gentle introduction, explaining the methods through examples taken from various fields of science.

As a computational scientist, you will work with other applications, models, and methods than those covered herein. The field is vast and it is impossible to capture more than a small fraction of it in a reasonably sized text. Therefore, we will teach principles and ideas. We believe that principles and ideas carry over from field to field, whereas particularly clever tricks tend to be application specific. We urge you to focus on the ideas and not to get too concerned about the context in which the models appear. We describe the context and provide examples merely to simplify the setting and make the text easier to read.

To read this text, you must know calculus (functions, differentiation, integration etc.) and the basics of linear algebra (vectors, matrices etc.), and you should be familiar with elementary programming. This book is just a gentle start to show what scientific computing is about and present some background that will simplify your future study of more advanced texts on numerical methods and their applications in science and engineering.

All the problems in the text have been solved, and the solutions are provided at

<http://www.ifi.uio.no/cs/>

where you can also find lecture slides for all the chapters.

We hope you enjoy reading this book as much as we have enjoyed writing it.

Fornebu
September 2010

*Aslak Tveito
Hans Petter Langtangen
Bjørn Fredrik Nielsen
Xing Cai*

Contents

1	Computing Integrals	1
1.1	Background	1
1.2	The Computational Problem	3
1.3	The Trapezoidal Method	4
1.3.1	Approximating the Integral Using One Trapezoid	5
1.3.2	Approximating the Integral Using Two Trapezoids	7
1.3.3	Approximating the Integral Using n Trapezoids	9
1.4	Computational Analysis	12
1.5	Back to the Bagels	14
1.6	Exercises	15
1.7	Projects	17
1.7.1	Show that the Error is $O(h^2)$	17
1.7.2	Derive Other Methods for Numerical Integration	22
1.7.3	Compute the Length of a Cable	26
2	Differential Equations: The First Steps	31
2.1	Modeling Changes	31
2.1.1	The Cultivation of Rabbits	32
2.1.2	The Simplest Possible Case	32
2.1.3	Exponential Growth	33
2.1.4	Logistic Growth	37
2.2	Numerical Solution	40
2.2.1	The Simplest Possible Model	41
2.2.2	Numerical Approximation of Exponential Growth	44
2.2.3	Numerical Stability	46
2.2.4	An Implicit Scheme	48
2.2.5	Explicit and Implicit Schemes	49
2.2.6	Numerical Solution of the Logistic Equation	55
2.3	Exercises	60
2.4	Projects	67
2.4.1	More on Stability	67
2.4.2	More on Accuracy	69

3	Systems of Ordinary Differential Equations	75
3.1	Rabbits and Foxes; Fish and Sharks	75
3.2	A Numerical Method: Unlimited Resources	78
3.3	A Numerical Method: Limited Resources	79
3.4	Phase Plane Analysis	80
3.4.1	A Simplified Model	80
3.4.2	The Phase Plane	81
3.4.3	Circles in the Analytical Phase Plane	83
3.4.4	Alternative Analysis	84
3.4.5	Circles in the Numerical Phase Plane	85
3.4.6	More on Numerics	88
3.5	Exercises	91
3.6	Project: Analysis of a Simple System	95
4	Nonlinear Algebraic Equations	99
4.1	The Bisection Method	102
4.2	Efficiency Consideration	106
4.3	Newton's Method	107
4.4	The Secant Method	112
4.5	Fixed-Point Iterations	115
4.5.1	Convergence of Fixed-Point Iterations	117
4.5.2	Speed of Convergence	120
4.5.3	Existence and Uniqueness of a Solution	121
4.5.4	Uniqueness	122
4.5.5	Existence	123
4.6	Systems of Nonlinear Equations	125
4.6.1	A Linear System	126
4.6.2	A Nonlinear System	127
4.6.3	Newton's Method	129
4.6.4	A Nonlinear Example	132
4.6.5	The Nonlinear System Revisited	133
4.7	Exercises	134
4.8	Project: Convergence of Newton's Method	138
5	The Method of Least Squares	147
5.1	Building Models of Discrete Data Sets	150
5.1.1	Approximation by a Constant	150
5.1.2	Approximation by a Linear Function	154
5.1.3	Approximation by a Quadratic Function	157
5.1.4	Large Data Sets	161
5.2	Application to Population Models	165
5.2.1	Exponential Growth of the World Population?	166
5.2.2	Logistic Growth of the World Population	170
5.3	Least Squares Approximations of Functions	173
5.3.1	Approximating Functions by a Constant	176

- 5.3.2 Approximation Using Linear Functions178
- 5.3.3 Approximation Using Quadratic Functions180
- 5.3.4 Summary of the Examples183
- 5.4 Exercises185
- 5.5 Project: Computing Coefficients190

- 6 About Scientific Software195**
 - 6.1 Algorithms Expressed as Pseudo Code197
 - 6.1.1 Basic Ingredients of Pseudo Codes197
 - 6.1.2 Integration199
 - 6.1.3 Optimization of Algorithms and Implementations200
 - 6.1.4 Developing Algorithms for Simpson’s Rule203
 - 6.1.5 Adaptive Integration Rules207
 - 6.1.6 Ordinary Differential Equations211
 - 6.2 About Programming Languages212
 - 6.2.1 Static Typing Versus Dynamic Typing213
 - 6.2.2 Computational Efficiency214
 - 6.2.3 Built-in High-Performance Utilities215
 - 6.2.4 Support for User-Defined Objects216
 - 6.3 The Trapezoidal Rule in Different Languages217
 - 6.3.1 Code Structure217
 - 6.3.2 Fortran 77218
 - 6.3.3 C and C++222
 - 6.3.4 Java224
 - 6.3.5 Matlab226
 - 6.3.6 Python228
 - 6.3.7 Vectorization230
 - 6.3.8 Maple236
 - 6.3.9 Summary237
 - 6.4 Heun’s Scheme in Different Languages239
 - 6.4.1 Code Structure239
 - 6.4.2 Fortran 77239
 - 6.4.3 Graphics242
 - 6.4.4 C++244
 - 6.4.5 Java248
 - 6.4.6 Matlab249
 - 6.4.7 Python251
 - 6.4.8 Summary252
 - 6.5 Numerical Software Engineering254
 - 6.5.1 Function Libraries254
 - 6.5.2 Motivation for Object-Oriented Libraries259
 - 6.5.3 Design of a Numerical Integration Hierarchy262
 - 6.5.4 A Class Hierarchy in Java263
 - 6.5.5 A Class Hierarchy in Python265
 - 6.5.6 Object-Oriented Programming in Matlab268

6.6	Exercises	269
6.7	Projects	270
6.7.1	Computing the Volume of a Cylindrical Container	270
6.7.2	A Class Hierarchy for Scalar ODE Solvers	271
6.7.3	Software for Systems of ODEs	272
7	The Diffusion Equation	273
7.1	Basics of Diffusion Processes	274
7.1.1	Heat Conduction	274
7.1.2	Diffusive Transport of a Substance	275
7.1.3	Diffusion Versus Convection	276
7.1.4	The Relevance of One-Dimensional Models	277
7.2	The Mathematical Model of Diffusion	285
7.2.1	The Diffusion Equation	285
7.2.2	Initial and Boundary Conditions	286
7.2.3	The One-Dimensional Initial-Boundary Value Problem ...	287
7.2.4	The Three-Dimensional Diffusion Equation	287
7.3	Derivation of Diffusion Equations	290
7.3.1	Diffusion of a Substance	290
7.3.2	Heat Conduction	295
7.3.3	Viscous Fluid Flow	300
7.3.4	Summarizing the Models	304
7.3.5	Scaling	307
7.4	Explicit Numerical Methods	316
7.4.1	The Basics of Finite Difference Discretizations	316
7.4.2	Incorporating Dirichlet Boundary Conditions	321
7.4.3	Incorporating Neumann Boundary Conditions	321
7.4.4	How to Verify a Computer Implementation	324
7.4.5	Instability	327
7.4.6	A Discrete Algorithm Directly from Physics	332
7.4.7	Variable Coefficients	334
7.5	Implicit Numerical Methods	336
7.5.1	The Backward Euler Scheme	336
7.5.2	The Linear System of Equations	337
7.5.3	Solution of Tridiagonal Linear Systems	340
7.5.4	Comparing the Explicit and Implicit Methods	342
7.5.5	The Crank–Nicolson Scheme	344
7.5.6	The θ Scheme	346
7.6	Exercises	348
7.7	Projects	352
7.7.1	Diffusion of a Jump	352
7.7.2	Periodical Injection of Pollution	353
7.7.3	Analyzing Discrete Solutions	355
7.7.4	Compare Different Scalings	356

- 8 Analysis of the Diffusion Equation**359
 - 8.1 Properties of the Solution360
 - 8.1.1 Energy Arguments360
 - 8.1.2 A Bound on the Derivative362
 - 8.1.3 Stability364
 - 8.1.4 Uniqueness366
 - 8.1.5 Maximum Principles366
 - 8.1.6 Physical Considerations367
 - 8.1.7 Analytical Considerations367
 - 8.1.8 The Minimum Principle372
 - 8.1.9 Summary373
 - 8.1.10 Uniqueness Revisited374
 - 8.2 Separation of Variables and Fourier Analysis374
 - 8.2.1 Separation of Variables375
 - 8.2.2 Super-Positioning378
 - 8.2.3 Fourier Series and the Initial Condition381
 - 8.2.4 Some Simple Examples381
 - 8.2.5 Initial Conditions Given by a Sum of Sine Functions382
 - 8.2.6 Computing Fourier Sine Series384
 - 8.2.7 Summary386
 - 8.2.8 More Examples387
 - 8.2.9 Analysis of an Explicit Finite Difference Scheme392
 - 8.2.10 Consequences of the Stability Criterion395
 - 8.2.11 Exercises396
 - 8.3 Projects403
 - 8.3.1 Neumann Boundary Conditions403
 - 8.3.2 Variable Coefficients406

- 9 Parameter Estimation and Inverse Problems**411
 - 9.1 Parameter Estimation in Exponential Growth412
 - 9.1.1 A Simpler Problem414
 - 9.2 The Backward Diffusion Equation415
 - 9.3 Estimating the Diffusion Coefficient418
 - 9.4 Exercises419

- 10 A Glimpse of Parallel Computing**423
 - 10.1 Motivations for Parallel Computing423
 - 10.1.1 From the Perspective of Speed423
 - 10.1.2 From the Perspective of Memory426
 - 10.1.3 Parallel Computers426
 - 10.2 More About Parallel Computing427
 - 10.2.1 Inspirations from Parallel Processing in Real Life428
 - 10.2.2 From Serial Computing to Parallel Computing428
 - 10.2.3 Example 1 of Data Parallelism430
 - 10.2.4 Example 2 of Data Parallelism432

- 10.2.5 Example 3 of Data Parallelism434
- 10.2.6 Performance Analysis438
- 10.2.7 Overhead Related to Parallelization440
- 10.3 Parallel Programming442
 - 10.3.1 OpenMP Programming443
 - 10.3.2 MPI Programming445
 - 10.3.3 Concluding Remarks449
- 10.4 Exercises449
- 10.5 Project: Parallel Solution of a 2D Diffusion Equation.....451

- References**455

- Index**457