

# Elements of Scientific Computing

A. Tveito, H. P. Langtangen, B. F. Nielsen and X. Cai

[ **simula** . research laboratory ]





## 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 flowering in many fields such as physics, mechanics, geophysics, astrophysics, chemistry, biology, and most engineering disciplines. The computational methods used in these branches of science and engineering are very similar. This book is a first introduction to such methods. Many books have been written about this subject. The present one had the overall goal of providing a gentle introduction where the methods are explained through examples taken from various fields of science.

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

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

All problems in the text have been solved and the solutions are provided on the following web page:

`http://www.ifi.uio.no/cs/`

There you can also find lecture slides for all the chapters in the text.

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

Fornebu, February 2009

Aslak Tveito<sup>1</sup>, Hans Petter Langtangen, Bjørn Fredrik Nielsen and Xing Cai

Simula Research Laboratory

Department of Informatics, University of Oslo

---

<sup>1</sup> This author has received financial support from the Norwegian non-fiction literature fund.



# Table of Contents

<b>1</b>	<b>Computing Integrals</b>	1
1.1	Background	1
1.2	The Computational Problem	2
1.3	The Trapezoidal Method	4
1.3.1	Approximating the Integral Using One Trapezoid	4
1.3.2	Approximating the Integral Using Two Trapezoids	7
1.3.3	Approximating the Integral Using $n$ Trapezoids	9
1.4	Computational Analysis	11
1.5	Back to the Bagels	13
1.6	Exercises	15
1.7	Projects	18
1.7.1	Show that the error is $O(h^2)$	18
1.7.2	Derive Other Methods for Numerical Integration	22
1.7.3	Compute the Length of a Cable	26
<b>2</b>	<b>Differential Equations; The First Steps</b>	31
2.1	Modeling Changes	31
2.1.1	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	45
2.2.4	An Implicit Scheme	47
2.2.5	Explicit and Implicit Schemes	48
2.2.6	Numerical Solution of the Logistic Equation	53
2.3	Exercises	59
2.4	Projects	66
2.4.1	More on Stability	66
2.4.2	More on Accuracy	68
<b>3</b>	<b>Systems of Ordinary Differential Equations</b>	73
3.1	Rabbits and Foxes; Fish and Sharks	73
3.2	A Numerical Method; Unlimited Resources	75
3.3	A Numerical Method; Limited Resources	76
3.4	Phase Plane Analysis	77
3.4.1	A Simplified Model	77
3.4.2	The Phase Plane	79

3.4.3	Circles in the Analytical Phase Plane . . . . .	79
3.4.4	Alternative Analysis . . . . .	81
3.4.5	Circles in the Numerical Phase Plane . . . . .	82
3.4.6	More on Numerics . . . . .	85
3.5	Exercises . . . . .	88
3.6	Projects . . . . .	92
3.6.1	Analysis of a Simple System . . . . .	92
<b>4</b>	<b>Nonlinear Algebraic Equations . . . . .</b>	<b>97</b>
4.1	The Bisection Method . . . . .	100
4.2	Efficiency Consideration . . . . .	103
4.3	Newton's Method . . . . .	105
4.4	The Secant Method . . . . .	110
4.5	Fixed-Point Iterations . . . . .	113
4.5.1	Convergence of Fixed-Point Iterations . . . . .	115
4.5.2	Speed of Convergence . . . . .	117
4.5.3	Existence and Uniqueness of a Solution . . . . .	118
4.5.4	Uniqueness . . . . .	119
4.5.5	Existence . . . . .	120
4.6	Systems of Nonlinear Equations . . . . .	122
4.6.1	A Linear System . . . . .	123
4.6.2	A Nonlinear System . . . . .	125
4.6.3	Newton's Method . . . . .	126
4.6.4	A Nonlinear Example . . . . .	128
4.6.5	The Nonlinear System Revisited . . . . .	129
4.7	Exercises . . . . .	130
4.8	Projects . . . . .	134
4.8.1	Convergence of Newton's Method . . . . .	134
<b>5</b>	<b>The Method of Least Squares . . . . .</b>	<b>141</b>
5.1	Making Models of Discrete Data Sets . . . . .	143
5.1.1	Approximation by a Constant . . . . .	144
5.1.2	Approximation by a Linear Function . . . . .	148
5.1.3	Approximation by a Quadratic Function . . . . .	152
5.1.4	Large Data Sets . . . . .	154
5.2	Application to Population Models . . . . .	159
5.2.1	Exponential Growth of the World Population? . . . . .	159
5.2.2	Logistic Growth of the World Population . . . . .	163
5.3	Least Squares Approximations of Functions . . . . .	166
5.3.1	Approximating Functions by a Constant . . . . .	167
5.3.2	Approximation Using Linear Functions . . . . .	170
5.3.3	Approximation Using Quadratic Functions . . . . .	173
5.3.4	Summary of the Examples . . . . .	175
5.4	Exercises . . . . .	177
5.5	Projects . . . . .	182

5.5.1	Computing Coefficients .....	182
<b>6</b>	<b>About Scientific Software .....</b>	<b>187</b>
6.1	Algorithms Expressed as Pseudo Code .....	189
6.1.1	Basic Ingredients of Pseudo Codes .....	189
6.1.2	Integration .....	191
6.1.3	Optimization of Algorithms and Implementations .....	192
6.1.4	Developing Algorithms for Simpson's Rule .....	195
6.1.5	Adaptive Integration Rules .....	199
6.1.6	Ordinary Differential Equations .....	203
6.2	About Programming Languages .....	205
6.2.1	Static Typing versus Dynamic Typing .....	206
6.2.2	Computational Efficiency .....	207
6.2.3	Built-in High-Performance Utilities .....	208
6.2.4	Support for User-Defined Objects .....	209
6.3	The Trapezoidal Rule in Different Languages .....	210
6.3.1	Structure of the Code .....	210
6.3.2	Fortran 77 .....	211
6.3.3	C and C++ .....	214
6.3.4	Java .....	217
6.3.5	Matlab .....	219
6.3.6	Python .....	222
6.3.7	Vectorization .....	224
6.3.8	Maple .....	229
6.3.9	Summary .....	231
6.4	Heun's Scheme in Different Languages .....	233
6.4.1	Structure of the Code .....	233
6.4.2	Fortran 77 .....	233
6.4.3	Graphics .....	236
6.4.4	C++ .....	239
6.4.5	Java .....	242
6.4.6	Matlab .....	243
6.4.7	Python .....	245
6.4.8	Summary .....	247
6.5	Numerical Software Engineering .....	248
6.5.1	Function Libraries .....	248
6.5.2	Motivation for Object-Oriented Libraries .....	254
6.5.3	Design of a Numerical Integration Hierarchy .....	256
6.5.4	A Class Hierarchy in Java .....	258
6.5.5	A Class Hierarchy in Python .....	260
6.5.6	Object-Oriented Programming in Matlab .....	263
6.6	Exercises .....	264
6.7	Projects .....	265
6.7.1	Computing the Volume of a Cylindrical Container .....	265
6.7.2	A Class Hierarchy for Scalar ODE Solvers .....	266

6.7.3	Software for Systems of ODEs . . . . .	267
<b>7</b>	<b>The Diffusion Equation . . . . .</b>	<b>269</b>
7.1	Basics of Diffusion Processes . . . . .	270
7.1.1	Heat Conduction . . . . .	270
7.1.2	Diffusive Transport of a Substance . . . . .	271
7.1.3	Diffusion versus Convection . . . . .	272
7.1.4	The Relevance of One-Dimensional Models . . . . .	273
7.2	The Mathematical Model of Diffusion . . . . .	283
7.2.1	The Diffusion Equation . . . . .	283
7.2.2	Initial and Boundary Conditions . . . . .	283
7.2.3	The One-Dimensional Initial-Boundary Value Problem . . . . .	284
7.2.4	The Three-Dimensional Diffusion Equation . . . . .	285
7.3	Derivation of Diffusion Equations . . . . .	287
7.3.1	Diffusion of a Substance . . . . .	287
7.3.2	Heat Conduction . . . . .	292
7.3.3	Viscous Fluid Flow . . . . .	297
7.3.4	Summarizing the Models . . . . .	300
7.3.5	Scaling . . . . .	303
7.4	Explicit Numerical Methods . . . . .	312
7.4.1	The Basics of Finite Difference Discretizations . . . . .	313
7.4.2	Incorporating Dirichlet Boundary Conditions . . . . .	317
7.4.3	Incorporating Neumann Boundary Conditions . . . . .	318
7.4.4	How to Verify a Computer Implementation . . . . .	320
7.4.5	Instability . . . . .	323
7.4.6	Discrete Algorithm Directly from Physics . . . . .	327
7.4.7	Variable Coefficients . . . . .	331
7.5	Implicit Numerical Methods . . . . .	333
7.5.1	The Backward Euler Scheme . . . . .	333
7.5.2	The Linear System of Equations . . . . .	334
7.5.3	Solution of Tridiagonal Linear Systems . . . . .	337
7.5.4	Comparing the Explicit and Implicit Methods . . . . .	340
7.5.5	The Crank-Nicolson Scheme . . . . .	341
7.5.6	The $\theta$ Scheme . . . . .	343
7.6	Exercises . . . . .	345
7.7	Projects . . . . .	350
7.7.1	Diffusion of a Jump . . . . .	350
7.7.2	Periodical Injection of Pollution . . . . .	351
7.7.3	Analysing Discrete Solutions . . . . .	352
7.7.4	Compare Different Scalings . . . . .	354



<b>8</b>	<b>Analysis of the Diffusion Equation</b>	355
8.1	Properties of the Solution	355
8.1.1	Energy Arguments	356
8.1.2	A Bound on the Derivative	358
8.1.3	Stability	360
8.1.4	Uniqueness	361
8.1.5	Maximum Principles	362
8.1.6	Physical Considerations	362
8.1.7	Analytical Considerations	363
8.1.8	The Minimum Principle	368
8.1.9	Summary	368
8.1.10	Uniqueness Revisited	369
8.2	Separation of Variables and Fourier Analysis	370
8.2.1	Separation of Variables	370
8.2.2	Super-Positioning	374
8.2.3	Fourier Series and the Initial Condition	376
8.2.4	Some Simple Examples	376
8.2.5	Initial Conditions Given by a Sum of Sine Functions	377
8.2.6	Computing Fourier Sine Series	379
8.2.7	Summary	381
8.2.8	More Examples	382
8.2.9	Analysis of an Explicit Finite Difference Scheme	388
8.2.10	Consequences of the Stability Criterion	391
8.2.11	Exercises	391
8.3	Projects	399
8.3.1	Neumann Boundary Conditions	399
8.3.2	Variable Coefficients	402
<b>9</b>	<b>Parameter Estimation and Inverse Problems</b>	405
9.1	Parameter Estimation in Exponential Growth	406
9.2	The Backwards Diffusion Equation	409
9.3	Estimating the Diffusion Coefficient	412
9.4	Exercises	413
<b>10 A</b>	<b>Glimpse of Parallel Computing</b>	415
10.1	Motivations for Parallel Computing	415
10.1.1	From the Perspective of Speed	415
10.1.2	From the Perspective of Memory	417
10.1.3	Parallel Computers	418
10.2	More About Parallel Computing	419
10.2.1	Inspirations from Parallel Processing in Real Life	419
10.2.2	From Serial Computing to Parallel Computing	420
10.2.3	Example 1 of Data Parallelism	422
10.2.4	Example 2 of Data Parallelism	424
10.2.5	Example 3 of Data Parallelism	426

10.2.6	Performance Analysis	430
10.2.7	Overhead Related to Parallelization	432
10.3	Parallel Programming	434
10.3.1	OpenMP Programming	435
10.3.2	MPI Programming	437
10.3.3	Concluding Remarks	441
10.4	Exercises	441
10.5	Project: Parallel Solution of a 2D Diffusion Equation	444