

Numerical Algorithms
and
Digital Representation

Knut Mørken
Department of Informatics
Centre of Mathematics for Applications
University of Oslo

August 2012

Preface

These lecture notes form part of the syllabus for the first-semester course MAT-INF1100 at the University of Oslo. The topics roughly cover two main areas: *Numerical algorithms*, and what can be termed *digital understanding*. Together with a thorough understanding of calculus and programming, this is knowledge that students in the mathematical sciences should gain as early as possible in their university career. As subjects such as physics, meteorology and statistics, as well as many parts of mathematics, become increasingly dependent on computer calculations, this training is essential.

Our aim is to train students who should not only be able *to use* a computer for mathematical calculations; they should also have a basic understanding of *how* the computational methods work. Such understanding is essential both in order to judge the quality of computational results, and in order to develop new computational methods when the need arises.

In these notes we cover the basic numerical algorithms such as interpolation, numerical root finding, differentiation and integration, as well as numerical solution of ordinary differential equations. In the area of digital understanding we discuss digital representation of numbers, text, sound and images. In particular, the basics of lossless compression algorithms with Huffman coding and arithmetic coding is included.

A basic assumption throughout the notes is that the reader either has attended a basic calculus course in advance or is attending such a course while studying this material. Basic familiarity with programming is also assumed. However, I have tried to quote theorems and other results on which the presentation rests. Provided you have an interest and curiosity in mathematics, it should therefore not be difficult to read most of the material with a good mathematics background from secondary school.

MAT-INF1100 is a central course in the project *Computers in Science Edu-*

cation (CSE) at the University of Oslo. The aim of this project is to make sure that students in the mathematical sciences get a unified introduction to computational methods as part of their undergraduate studies. The basic foundation is laid in the first semester with the calculus course, MAT1100, and the programming course INF1100, together with MAT-INF1100. The maths courses that follow continue in the same direction and discuss a number of numerical algorithms in linear algebra and related areas, as well as applications such as image compression and ranking of web pages.

Some fear that a thorough introduction of computational techniques in the mathematics curriculum will reduce the students' basic mathematical abilities. This could easily be true if the use of computations only amounted to running code written by others. However, deriving the central algorithms, programming them, and studying their convergence properties, should lead to a level of mathematical understanding that should certainly match that of a more traditional approach.

Many people have helped develop these notes which have matured over a period of ten years. Øyvind Ryan, Andreas Våvang Solbrå, Solveig Bruvoll, and Marit Sandstad have helped directly with recent versions, while Pål Hermunn Johansen provided extensive programming help with an earlier version. For this latest version, Andreas Våvang Solbrå has provided important assistance and feedback. Geir Pedersen was my co-lecturer for four years. He was an extremely good discussion partner on all the facets of this material, and influenced the list of contents in several ways. I work at the Centre of Mathematics for Applications (CMA) at the University of Oslo, and I am grateful to the director, Ragnar Winther, for his enthusiastic support of the CSE project and my extensive undertakings in teaching. Over many years, my closest colleagues Geir Dahl, Michael Floater, and Tom Lyche have shaped my understanding of numerical analysis and allowed me to spend considerably more time than usual on elementary teaching. Another colleague, Sverre Holm, has been my source of information on signal processing. To all of you: thank you!

My academic home, the Department of Informatics and its chairman Morten Dæhlen, has been very supportive of this work by giving me the freedom to extend the Department's teaching duties, and by extensive support of the CSE-project. It has been a pleasure to work with the Department of Mathematics over the past eight years, and I have many times been amazed by how much confidence they seem to have in me. I have learnt a lot, and have thoroughly enjoyed teaching at the cross-section between mathematics and computing which is my scientific home. I can only say thank you, and I feel at home in both departments.

A course like MAT-INF1100 is completely dependent on support from other

courses. Tom Lindstrøm has done a tremendous job with the parallel calculus course MAT1100, and its sequel MAT1110 on multivariate analysis and linear algebra. Hans Petter Langtangen has done an equally impressive job with INF1100, the introductory programming course with a mathematical and scientific flavour, and I have benefited from many hours of discussions with both of them. Morten Hjorth-Jensen, Arnt-Inge Vistnes and Anders Malthe-Sørenssen with colleagues have introduced a computational perspective in a number of physics courses, and discussions with them have convinced me of the importance of introducing computations for all students in the mathematical sciences. Thank you to all of you.

The CSE project is run by a group of people: Morten Hjorth-Jensen and Anders Malthe-Sørenssen from the Physics Department, Hans Petter Langtangen from the Simula Research Lab and the Department of Informatics, Øyvind Ryan from the CMA, Annik Myhre (Dean of Education at the MN-faculty¹), Hanne Sølna (Head of the Studies section at the MN-faculty¹), Helge Galdal (Administrative Leader of the CMA), and myself. This group of people has been the main source of inspiration for this work, and without you, there would still only be uncoordinated attempts at including computations in our elementary courses. Thank you for all the fun we have had.

The CSE project has become much more than I could ever imagine, and the reason is that there seems to be a genuine collegial atmosphere at the University of Oslo in the mathematical sciences. This means that it has been possible to build momentum in a common direction not only within a research group, but across several departments, which seems to be quite unusual in the academic world. Everybody involved in the CSE project is responsible for this, and I can only thank you all.

Finally, as in all teaching endeavours, the main source of inspiration is the students, without whom there would be no teaching. Many students become frustrated when their understanding of the nature of mathematics is challenged, but the joy of seeing the excitement in their eyes when they understand something new is a constant source of satisfaction.

Blindern, August 2012

Knut Mørken

¹The Faculty of Mathematics and Natural Sciences.

Contents

1 Introduction	1
1.1 A bit of history	1
1.2 Computers and different types of information	3
1.2.1 Text	4
1.2.2 Sound	4
1.2.3 Images	4
1.2.4 Film	4
1.2.5 Geometric form	5
1.2.6 Laws of nature	5
1.2.7 Virtual worlds	5
1.2.8 Summary	6
1.3 Computation by hand and by computer	7
1.4 Algorithms	8
1.4.1 Statements	10
1.4.2 Variables and assignment	11
1.4.3 For-loops	12
1.4.4 If-tests	13
1.4.5 While-loops	14
1.4.6 Print statement	14
1.5 Doing computations on a computer	15
1.5.1 How can computers be used for calculations?	15
1.5.2 What do you need to know?	16
1.5.3 Different computing environments	17
1.5.4 Exercises	19

I	Numbers	21
2	0 and 1	23
2.1	Robust communication	23
2.2	Why 0 and 1 in computers?	24
2.3	True and False	25
2.3.1	Logical variables and logical operators	26
2.3.2	Combinations of logical operators	29
2.3.3	Exercises	30
3	Numbers and Numeral Systems	31
3.1	Terminology and Notation	31
3.1.1	Exercises	33
3.2	Natural Numbers in Different Numeral Systems	34
3.2.1	Alternative Numeral Systems	34
3.2.2	Conversion to the Base- β Numeral System	37
3.2.3	Conversion between base-2 and base-16	39
3.2.4	Exercises	41
3.3	Representation of Fractional Numbers	43
3.3.1	Rational and Irrational Numbers in Base- β	43
3.3.2	An Algorithm for Converting Fractional Numbers	47
3.3.3	Conversion between binary and hexadecimal	48
3.3.4	Properties of Fractional Numbers in Base- β	49
3.3.5	Exercises	52
3.4	Arithmetic in Base β	54
3.4.1	Addition	54
3.4.2	Subtraction	54
3.4.3	Multiplication	55
3.4.4	Exercises	55
4	Computers, Numbers, and Text	59
4.1	Representation of Integers	59
4.1.1	Bits, bytes and numbers	60
4.1.2	Fixed size integers	61
4.1.3	Two's complement	62
4.1.4	Integers in Java	63
4.1.5	Integers in Python	64
4.1.6	Division by zero	65
4.1.7	Exercises	65
4.2	Computers and real numbers	66

4.2.1	The challenge of real numbers	66
4.2.2	The normal form of real numbers	67
4.2.3	32-bit floating-point numbers	69
4.2.4	Special bit combinations	70
4.2.5	64-bit floating-point numbers	70
4.2.6	Floating point numbers in Java	71
4.2.7	Floating point numbers in Python	71
4.2.8	Exercises	71
4.3	Representation of letters and other characters	72
4.3.1	The ASCII table	73
4.3.2	ISO latin character sets	75
4.3.3	Unicode	76
4.3.4	UTF-8	78
4.3.5	UTF-16	80
4.3.6	UTF-32	82
4.3.7	Text in Java	82
4.3.8	Text in Python	82
4.3.9	Exercises	83
4.4	Representation of general information	86
4.4.1	Text	86
4.4.2	Numbers	87
4.4.3	General information	88
4.4.4	Computer programs	88
4.5	A fundamental principle of computing	88
4.5.1	Exercises	90
5	Computer Arithmetic and Round-Off Errors	91
5.1	Integer arithmetic and errors	92
5.2	Floating-point arithmetic and round-off error	92
5.2.1	Truncation and rounding	93
5.2.2	A simplified model for computer arithmetic	94
5.2.3	An algorithm for floating-point addition	95
5.2.4	Observations on round-off errors in addition/subtraction	97
5.2.5	Multiplication and division of floating-point numbers	99
5.2.6	The IEEE standard for floating-point arithmetic	101
5.2.7	Exercises	101
5.3	Measuring the error	103
5.3.1	Absolute error	104
5.3.2	Relative error	104
5.3.3	Properties of the relative error	105

5.3.4	Errors in floating-point representation	107
5.3.5	Exercises	108
5.4	Rewriting formulas to avoid rounding errors	110
5.4.1	Exercises	111

II Sequences of Numbers 115

6 Numerical Simulation of Difference Equations 117

6.1	Why equations?	117
6.1.1	Exercises	118
6.2	Difference equations defined	118
6.2.1	Initial conditions	121
6.2.2	Linear difference equations	122
6.2.3	Solving difference equations	122
6.2.4	Exercises	123
6.3	Simulating difference equations	124
6.3.1	Exercises	127
6.4	Review of the theory for linear equations	128
6.4.1	First-order homogenous equations	128
6.4.2	Second-order homogenous equations	129
6.4.3	Linear homogenous equations of general order	132
6.4.4	Inhomogenous equations	133
6.4.5	Exercises	136
6.5	Simulation of difference equations and round-off errors	137
6.5.1	Explanation of example 6.27	140
6.5.2	Round-off errors for linear equations of general order	143
6.5.3	Exercises	144
6.6	Summary	147

7 Lossless Compression 149

7.1	Introduction	150
7.1.1	Run-length coding	151
7.1.2	Exercises	151
7.2	Huffman coding	152
7.2.1	Binary trees	154
7.2.2	Huffman trees	155
7.2.3	The Huffman algorithm	156
7.2.4	Properties of Huffman trees	160

7.2.5	Exercises	161
7.3	Probabilities and information entropy	162
7.3.1	Probabilities rather than frequencies	162
7.3.2	Information entropy	164
7.3.3	Exercises	165
7.4	Arithmetic coding	166
7.4.1	Arithmetic coding basics	166
7.4.2	An algorithm for arithmetic coding	168
7.4.3	Properties of arithmetic coding	172
7.4.4	A decoding algorithm	174
7.4.5	Arithmetic coding in practice	176
7.4.6	Exercises	177
7.5	Lempel-Ziv-Welch algorithm	178
7.6	Lossless compression programs	179
7.6.1	Compress	179
7.6.2	gzip	179
7.6.3	Exercises	179
8	Digital Sound	181
8.1	Sound	181
8.1.1	Loudness: Sound pressure and decibels	182
8.1.2	The pitch of a sound	185
8.1.3	Any function is a sum of sin and cos	187
8.1.4	Exercises	190
8.2	Digital sound	190
8.2.1	Sampling	190
8.2.2	Limitations of digital audio: The sampling theorem	191
8.2.3	Reconstructing the original signal	193
8.2.4	Exercises	194
8.3	Simple operations on digital sound	195
8.4	More advanced sound processing	200
8.4.1	The Discrete Cosine Transform	200
8.5	Lossy compression of digital sound	201
8.6	Psycho-acoustic models	204
8.7	Digital audio formats	205
8.7.1	Audio sampling— PCM	205
8.7.2	Lossless formats	206
8.7.3	Lossy formats	207

9 Polynomial Interpolation	211
9.1 The Taylor polynomial with remainder	212
9.1.1 The Taylor polynomial	212
9.1.2 The remainder	214
9.1.3 Exercises	219
9.2 Interpolation and the Newton form	221
9.2.1 The interpolation problem	221
9.2.2 The Newton form of the interpolating polynomial	223
9.2.3 Exercises	226
9.3 Divided differences	228
9.3.1 Exercises	231
9.4 Computing with the Newton form	233
9.4.1 Exercises	236
9.5 Interpolation error	236
9.6 Summary	238
10 Zeros of Functions	239
10.1 The need for numerical root finding	240
10.1.1 Analysing difference equations	240
10.1.2 Labelling plots	241
10.2 The Bisection method	242
10.2.1 The intermediate value theorem	242
10.2.2 Derivation of the Bisection method	243
10.2.3 Error analysis	246
10.2.4 Revised algorithm	248
10.2.5 Exercises	250
10.3 The Secant method	251
10.3.1 Basic idea	252
10.3.2 Testing for convergence	254
10.3.3 Revised algorithm	255
10.3.4 Convergence and convergence order of the Secant method	256
10.3.5 Exercises	258
10.4 Newton's method	258
10.4.1 Basic idea	259
10.4.2 Algorithm	259
10.4.3 Convergence and convergence order	261
10.4.4 Exercises	263
10.5 Summary	266

11 Numerical Differentiation	269
11.1 Newton's difference quotient	270
11.1.1 The basic idea	271
11.1.2 The truncation error	272
11.1.3 The round-off error	274
11.1.4 Optimal choice of h	280
11.1.5 Exercises	281
11.2 Summary of the general strategy	284
11.2.1 Exercises	285
11.3 A symmetric version of Newton's quotient	285
11.3.1 Derivation of the method	285
11.3.2 The error	287
11.3.3 Optimal choice of h	289
11.3.4 Exercises	290
11.4 A four-point differentiation method	291
11.4.1 Derivation of the method	292
11.4.2 The error	292
11.4.3 Exercises	294
11.5 Numerical approximation of the second derivative	295
11.5.1 Derivation of the method	295
11.5.2 The error	295
11.5.3 Optimal choice of h	297
11.5.4 Exercises	297
12 Numerical Integration	301
12.1 General background on integration	302
12.1.1 Exercises	305
12.2 The midpoint rule for numerical integration	305
12.2.1 A detailed algorithm	306
12.2.2 The error	308
12.2.3 Estimating the step length	312
12.2.4 Exercises	312
12.3 The trapezoidal rule	313
12.3.1 The error	315
12.3.2 Exercises	318
12.4 Simpson's rule	319
12.4.1 Derivation of Simpson's rule	319
12.4.2 Composite Simpson's rule	321
12.4.3 The error	323
12.4.4 Exercises	324

12.5 Summary	326
13 Numerical Solution of Differential Equations	327
13.1 What are differential equations?	327
13.1.1 An example from physics	327
13.1.2 General use of differential equations	329
13.1.3 Different types of differential equations	330
13.1.4 Exercises	330
13.2 First order differential equations	331
13.2.1 Initial conditions	331
13.2.2 A geometric interpretation of first order differential equations	333
13.2.3 Conditions that guarantee existence of one solution	335
13.2.4 What is a numerical solution of a differential equation?	336
13.2.5 Exercises	336
13.3 Euler's method	337
13.3.1 Basic idea and algorithm	337
13.3.2 Geometric interpretation	340
13.3.3 Exercises	341
13.4 Error analysis for Euler's method	343
13.4.1 Round-off error	344
13.4.2 Local and global error	344
13.4.3 Untangling the local errors	346
13.4.4 Exercises	348
13.5 Differentiating the differential equation	349
13.5.1 Exercises	351
13.6 Taylor methods	352
13.6.1 The quadratic Taylor method	352
13.6.2 Taylor methods of higher degree	355
13.6.3 Error in Taylor methods	355
13.6.4 Exercises	356
13.7 Midpoint Euler and other Runge-Kutta methods	357
13.7.1 Euler's midpoint method	358
13.7.2 The error	360
13.7.3 Runge-Kutta methods	360
13.7.4 Exercises	362
13.8 Systems of differential equations	365
13.8.1 Vector notation and existence of solution	365
13.8.2 Numerical methods for systems of first order equations	367
13.8.3 Higher order equations as systems of first order equations	369
13.8.4 Exercises	372

13.9 Final comments	372
13.9.1 Exercises	373

III Functions of two variables 377

14 Functions of two variables 379

14.1 Basics	379
14.1.1 Basic definitions	379
14.1.2 Differentiation	381
14.1.3 Vector functions of several variables	383
14.1.4 Exercises	384
14.2 Numerical differentiation	384
14.2.1 Exercises	388

15 Digital images and image formats 389

15.1 What is an image?	389
15.1.1 Light	389
15.1.2 Digital output media	390
15.1.3 Digital input media	391
15.1.4 Definition of digital image	391
15.1.5 Images as surfaces	393
15.1.6 Exercises	395
15.2 Operations on images	395
15.2.1 Normalising the intensities	396
15.2.2 Extracting the different colours	396
15.2.3 Converting from colour to grey-level	397
15.2.4 Computing the negative image	399
15.2.5 Increasing the contrast	399
15.2.6 Smoothing an image	401
15.2.7 Detecting edges	402
15.2.8 Comparing the first derivatives	406
15.2.9 Second-order derivatives	406
15.2.10 Exercises	408
15.3 Image formats	408
15.3.1 Raster graphics and vector graphics	408
15.3.2 Vector graphics formats	410
15.3.3 Raster graphics formats	411
15.3.4 Exercises	413

Answers 415

